国外经典教材

Classical Texts From Top Universities

Discrete Mathematics and Its Applications (Fourth Edition)

离散数学及其应用

（原书第4版）

（美） Kenneth H. Rosen 著

袁崇义 屈婉玲 王捍贫 刘田 译



机械工业出版社
China Machine Press



Education

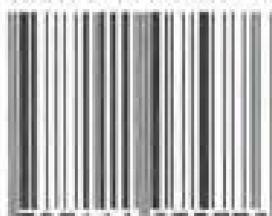
国外经典教材 Classical Texts From Top Universities

离散数学及其应用

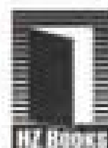
（原书第4版）

Discrete Mathematics and Its Applications

(Fourth Edition)



Mc Graw Hill



华章图书

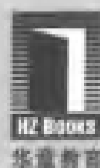
www.china-pub.com

北京市西城区百万庄南街1号 100037

购书热线：(010)68995259，8006100280（北京地区）

ISBN 7-111-07577-3/TP · 1205

定价：75.00 元



北京华章图文信息有限公司

国外经典教材



Classical Texts From Top Universities

（原书第4版）

离散数学及其应用

Discrete Mathematics and Its Applications

(Fourth Edition)

（美）Kenneth H. Rosen 著

袁崇义 屈婉玲 王捍贫 刘田 译



机械工业出版社
China Machine Press

本书介绍了离散数学的理论和方法，内容涉及数学推理、组合分析、离散结构和算法设计。本书取材极其广泛，除包括定义、定理的严密陈述外，还配备大量的实例和图、表的说明，适合各种需求的练习和题目，以及丰富的历史资料和网站资源。本书的第3版曾被全世界几百所大学选为教材，第4版作了新的改进和补充。本书适合于数学、计算机科学和工程技术专业人员使用。

本书版权登记号：图字：01-1999-2594

图书在版编目（CIP）数据

离散数学及其应用/（美）罗森（Rosen, K. H.）著；袁崇义，屈婉玲等译. -北京：机械工业出版社，2002.1
（国外经典教材）
书名原文：Discrete Mathematics and Its Applications, Fourth Edition
ISBN 7-111-07577-3

Ⅰ.离… Ⅱ.①罗…②袁…③屈… Ⅲ.离散数学 Ⅳ.O158

中国版本图书馆CIP数据核字（2001）第038005号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：马 珂
北京市密云县印刷厂印刷·新华书店北京发行所发行
2002年1月第1版第1次印刷
787mm×1092mm 1/16·52印张
印数：0 001-5 000册
定价：75.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭櫫了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自 1998 年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与 Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann 等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出 Tanenbaum，Stroustrup，Kernighan，Jim Gray 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学，西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“国外经典教材”是响应教育部提出的使用外版教材的号召，为国内高校的计算机本科教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这 20 多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被 M. I. T.，Stanford，U. C. Berkley，C. M. U. 等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：(010) 68995265
联系地址：北京市西城区百万庄南街 1 号
邮政编码：100037

专家指导委员会

（按姓名笔画顺序）

译 者 序

离散数学既是计算机科学的理论基础，又是计算机应用必不可少的工具。《离散数学及其应用》的写作目标就是向读者展示离散数学的实用性：为计算机专业学生提供一切必要的数学基础，使数学专业学生理解数学概念的重要性以及这些概念为什么对应用而言是重要的。作者肯尼斯 H. 罗森博士具有很深的数学造诣，有丰富的教学经验。我们翻译的这一本书是该书的第 4 版，其第 3 版曾在欧美 400 多所学校使用，获得了很大的成功。成功的作品还要修订，还要出新版，可见作者对学术的追求和对读者的热爱。

新版并不是简单纠正旧版的错误。信息时代使作者有可能与广大读者，包括教师、学生和自学者，保持及时的联系。新版对原版的修正反映的正是来自教和学两种实践的宝贵意见。新增加的内容既包括离散数学新的研究成果，也包括网络时代新的应用实例。

本书供 1 至 2 个学期使用。本书使学生学会特定的一些数学事实并知道怎样应用，更重要的是教会学生做数学思维，因此本书强调数学推理和用不同的方法解题。

本书有五个重要的主题融为一体：数学推理、组合分析、离散结构、算法思维以及应用和模拟。一门成功的离散数学课应该使这五部分内容有机地交融和取得平衡。

阅读、理解和构造数学证明依赖于数学推理的能力。逻辑和数学归纳技术是数学推理的基础。组合分析是一项重要的解题技巧，它指的是借助枚举而非公式来解决数学问题。离散结构是抽象的数学结构，包括集合、置换、关系、图、树和有限状态机等，用于表示离散对象及离散对象之间的关系。借助计算机解题需要给出程序，程序设计必须从算法规范入手。算法思维指的是算法设计（给出规范）、正确性证明和复杂性分析。离散数学已应用到包括化学、植物学、动物学、语言学、地理学、商业以及互联网等众多的领域，为各个领域建立数学模型是离散数学应用的基础。应用模拟是另一项重要的解题技巧。

显然，以上五个主题互为依存，它们构成离散数学的整体。本书既说明了它们独立的作用，又突出了它们有机的结合。

罗森博士在写作过程中充分考虑了教和学两个方面的需要，为教师和学生分别编撰了参考资料，并在书中提供了各章节相关的网站地址。从这些方面来看，这是译者见过的最完备、最成熟的教材。

译者大多有多年的从事计算机专业离散数学课程教学的经验，他们参与编撰的离散数学教材获得过多种奖励。希望本译著的出版能为我国离散数学的教与学提供宝贵的参考。

直接参与翻译的共有 4 人：袁崇义、屈婉玲、王捍贫、刘田。

译 者

2001 年 4 月

译 者 简 介

袁崇义　男，1941 年生，山东邹平人；1964 年毕业于南京大学数学系；现为北京大学计算机系教授，博士生导师，计算机理论教研室主任；研究兴趣包括并行处理的形式化方法和 Petri 网。

屈婉玲　女，1970 年毕业于北京大学物理系；现为北京大学计算机系教授，副系主任；研究兴趣包括并行算法、并行理论。

王捍贫　男，1993 年于北京师范大学数学系获博士学位；现为北京大学计算机系副教授；研究兴趣包括数理逻辑、形式语义和程序正确性证明。

刘田　1966 年生；1989 年从中国科技大学数学系本科毕业，1999 年在北京大学计算机系获博士学位；主要从事离散数学教学和计算复杂性理论研究。

前　言

多年来教授离散数学的经验和兴趣指引我写作本书。对学生而言，我的目的是为他们提供准确而可读的教材，使离散数学的概念和技术得以清晰地介绍和演示，我的目标是向爱怀疑的学生们展示离散数学的相关领域和实用性。我希望为学习计算机科学的学生提供一切必需的数学基础。我希望使学数学的学生理解数学概念的重要性以及这些概念为什么对应用而言是重要的。而且我希望既能达到这些目标，又不使教材含太多的水分。

对教师而言，我的目的是使用成熟的数学教学技术设计一个灵活而全面的教学工具。我希望为教师们提供一套有效的教材，使他们能高效地以适合于特定学生特点的方式教授离散数学。我希望已经达到这些目标。

我为此教材已经取得的巨大成功而分外高兴。此次第 4 版的许多改进都是成功使用本书的 400 多所学校大批师生反馈和建议的结果。此版有许多提高之处。原有的辅助材料更加丰富，还有配套网站提供的辅助材料，使它更易于被师生使用以达到他们的目标。

本教材为是 1 至 2 个学期的入门离散数学课而写的，适用于包括数学专业、计算机科学专业和工程专业在内的许多专业的学生。大学代数是它唯一的预备课程。

离散数学课的目标

一门离散数学课有多个目标。学生应该学会特定的一些数学事实并知道怎样应用；更重要的是，这样一门课应教会学生怎样作数学思维。为达到这些目标，本教材强调数学推理及用不同的方法解题。本教材有 5 个重要的主题交织在一起：数学推理、组合分析、离散结构、算法思考以及应用和建模。一门成功的离散数学课应该细心地使这五部分内容交融和取得平衡。

数学推理：学生必须理解数学推理以便阅读、理解和构造数学证明。本教材以数理逻辑开篇，因为数理逻辑是随后讨论的证明方法的基础。数学归纳技术是通过许多例子来重点介绍的。通过这些例子还仔细地说明了为什么数学归纳是有效的证明技术。

组合分析：解题的一项重要技巧是计数或枚举对象的能力，本书中对枚举的讨论就从基本的计数技术着手。重点是用组合分析来解决计数问题而不使用公式。

离散结构：一门离散数学课应该教学生如何使用离散结构，离散结构是抽象的数学结构，用来表示离散对象及离散对象之间的关系。离散结构包括集合、置换、关系、图、树和有限状态机。

算法思考：有几类问题是从给出算法说明入手求解的。描述了算法以后就可构造计算机程序来实现它，这一过程中的数学部分包括算法说明，证实它能正确执行，以及分析执行这一算法所需要的计算机内存和时间。所有这些内容均在本书中介绍。算法是用文字陈述和易于理解的一种伪码这样两种方式描述的。

应用与建模：离散数学已被应用到几乎所有研究领域。本书既有许多计算机科学和数据网络的应用实例，也有各式各样领域中的应用实例，包括化学、植物学、动物学、语言学、

地理、商业以及因特网。这些实例均是离散数学的自然而重要的应用，不是编造的。用离散数学建模是十分重要的解题技巧，本书的练习使学生有机会通过构造自己的模型来发展这一技巧。

为什么要出第4版

本书第3版在美国的400多所学校，加拿大的几十所大学，以及在欧洲、亚洲和大洋洲的大学使用获得成功。许多学生和教授均喜欢第3版的形式，那么为什么还要出第4版？这个问题值得认真回答。

首先，尽管第3版使用起来十分有效，许多教师还是要求做某些特定的改进。许多人希望改动正文，增加例题或使例题更易于理解，增加某种类型的练习，或增加能覆盖新内容的练习等。在新版中我根据已收到的大量建议对本书作了改进。根据用户要求做的改动使本版变得更好。

第二，离散数学是一个活跃的学科，每年都有许多新发现，其中有一些可以反映在教科书中。于是我在本版中收入了自第3版以来的某些新发现（随后的新发现将在本版以后印刷时尽可能收入，在网站上也会有反映）。

第三，自第3版发行以来，因特网变得十分重要，十分有用。在本版中将有把离散数学的应用和因特网自身结构联系起来的例题和练习。与本版配套的还有一个内容广泛的网站，它能对正文作有益的补充，为师生提供额外材料。想更多学习离散数学的人还可以通过本网站提供的路径访问网上有关的网站。不过，由于许多人不选用与本课程相连的网站，所以本书正文中给出了若干 Web 图标，指示网站链接，作为注释本书网站的网上指南。

下面列出的是本版中为使本书更有效所做的主要变动。

1. *新添内容*
   - 除大 O 记号以外新增加了大 Ω 和大 Θ 记号。
   - 概率论的新内容包括随机变量的方差和切比雪夫不等式。此外，还对 Monty 大厅三门问题做了讨论。
   - 对停机问题做了处理，包括其不可解的证明。
   - 讨论了流动推销员问题。
2. *扩充了原有内容*
   - 增加了关于数理逻辑和数学推理的附加材料，用新增例题说明怎样在量化语句和文字陈述之间互译，强化了推理规则的讨论。特别是关于量化语句的推理规则，现在明确地做了讨论，并且增加了阐明为何使用推理规则的例题。
   - 加强了对底函数和顶函数的讨论。
   - 正文中现在专有一节讨论生成函数，这是对原书附录内容的扩充。这一节的中心是生成函数怎样用于解决计数问题，解决递归关系，及证明组合恒等式。
   - 常系数非齐次线性递归关系现在在正文中讨论，而不是放在一组练习题中。
   - 对整数序列进行了更多的介绍，增加了从初始项识别整数序列可能的通项公式的例题和练习。
   - 增加了传记内容，包括 Peirce(皮尔斯)、Chebyshev(切比雪夫)、Knuth(克努斯)、Hardy(哈弟)、Ramanujan(拉曼扭因)、Tukey(图凯)、Sloane(斯罗尼)和 Mersenne(莫孙尼)。

3. 跟上时代的新例题

- 在课文某些关键之处增加了例题，用以帮助解释学生难于理解的重要概念，使课本更有趣。
- 增加了说明离散数学应用于因特网通信协议和网络结构的例题和练习。其中包括：与因特网地址及因特网协议包有关的计数问题；因特网搜索引擎使用的布尔搜索问题；还增加了一个关于怎样在 IP 组播中使用生成树的例题。
- 增加了材料以说明离散数学仍有许多未解决的问题，仍是不断有新发现的活跃领域。例如增加了莫孙尼素数的内容，包括 1997 年和 1998 年新发现的素数，还讨论了哥德巴赫猜想已经证明有效的范围，阐述了汉诺塔难题的变种，即有四根塔柱的问题。

4. 扩充了练习题

- 根据使用第 3 版的教师们的要求，增加了 500 多道练习题，包括常规的和有挑战性的问题，还包括基于逻辑和数学智力游戏的练习。新的分块练习题以一系列步骤展开一些关键概念。新练习能保证所有重要类型的题目既有奇数编号的，也有偶数编号的。此外，还有与以前学过的微积分有关的练习，不过对这些习题按习惯作了注明，不想做的话很容易就能避开它们。

5. 网上支持

- 做为正文的补充，已建立了一个既适合于学生也适合于教师的网站，它包含范围广泛的内容（见“配套网站”），包括一个注释性的网上指南，列出因特网上相关的网站。这一指南对课本正文能提供重要线索，在本版整个生命期中将不断更新以保证跟上发展。
- 课文中凡是网上指南有链接指向与所讨论内容相关的网站的地方，均有 Web 图标。(指南中不同的链接有 200 多个。) 网上的这些网站包含有关概念和应用的补充材料、名人传记、最新发现、可下载的源代码、交互式小应用程序（applet）、动画的算法以及其他有趣的内容。

特别之处

易入门　实践证明本书对初学者来说易读易懂。它的大部分内容只要求学生学过大学代数，不需要其他的预备知识，少数涉及微积分的地方均有明确的说明。大部分学生应该很容易理解课文中用于表示算法的伪码，不管他们是否学过程序设计语言。本书不要求形式化的计算机科学方面的预备知识。

灵活　本书为灵活使用作了精心设计。各章对其前面内容的依赖降到最小。每一章都分成长度大致相等的若干节，每一节又根据内容的自然分块划分成小节以便教学。教师可以根据这些分块很容易地安排进度。

写作风格　本书体现的写作风格是直接和实用。使用了准确的数学语言，但没有过份的形式化与过份的抽象，在适当的地方引入并使用记号。在数学陈述中对记号和文字的平衡作了仔细的考虑。

广泛的课堂实践　本书已在 400 多所学校使用过，其中 325 所以上使用了不止一次。来自许多学校师生的反馈使第 4 版成为比前几版更成功的教学工具。

数学上的严密性和准确性　本书所有的定义和定理均是分外细心陈述的。所以学生可以

欣赏其语言的准确以及数学上的严密性。证明则是缓慢引入并展开的，每一步都经细心论证，递归定义均有周密的解释并大量使用。

图和表 本书有550多幅图，用于阐明关键的概念和证明步骤。图上带有细心选用的颜色用以解释重点。只要可能，就用表格来小结关键内容或说明量化关系。

实例 本书用650多个实例阐明概念，表示不同内容的关系，以及引入应用。在例题中，首先提出一个问题，再按适度的细节给出它的解。

应用 本书包含的应用展示了离散数学在解决现实世界问题中的使用价值。本书所含的应用涉及的范围很广，包括计算机科学、数据网络、心理学、化学、工程、语言学、生物学、商业和因特网。

算法 离散数学的结果常能用算法表示，因此本书每一章均介绍了关键算法。这些算法一方面用文字描述给出，另一方面又用一种易于理解的结构化伪码的形式给出。附录2中对伪码作了描述和说明。书中对算法的计算复杂性也有初步的分析。

历史资料 本书对许多题材的背景作了简要的介绍。书中以脚注的形式给出了55位以上的数学家和计算机科学家的传记。传记中介绍了对离散数学作出巨大贡献的这些科学家们的生活、事业及成就。此外，作为对正文中历史资料的补充，还有大量史实的脚注。

关键术语和结果 每一章后面都列出了本章的关键术语和结果。关键术语只包括学生必须学会的那些最重要的而不是该章中定义的所有术语。

练习 正文中有3000多道练习，提出了大量不同类型的问题。有足够多的简单直接的问题用于开发基本技巧，还有大量的中等程度的练习和许多有挑战性的练习。练习的叙述是明白而无二义性的，全部按难易程度分级。分组给出的练习包含专门的讨论，提出正文中没有涉及的新概念，使学生可以通过自己的努力发现新思想。

比平均水平稍难的练习都用一个星号作了标记；相当有挑战性的问题则用两个星号标记。必须用微积分来解的练习均有明确说明。能导出正文中用到的结果的练习则用手指符号☞指明，在正文最后给出了所有奇数号的练习的答案或解题概略，其中包括清楚给出的大多数证明步骤。

复习题 每章最后都有一组复习题。这些问题的设计目的是帮助学生集中学习该章最重要的概念和技术。学生必须写出长长的答案才能回答这些问题，而不能只作计算或简短的应答。

补充练习 每章后面都有一组丰富而多变的补充练习。这些练习一般来说都比每一节后面的练习难度大。补充练习使一章的概念得以加强，并把不同内容更好地融为一体。

计算机题目 每一章后面还有一组计算机题目。这大约150个计算机题目可以把学生已经学到的计算和离散数学的内容联系起来。从数学角度或程序设计角度看来难度超过平均水平的计算机题目用一个星号标记，特别难的则用两个星号标记。

计算和研究 每一章的结尾都有一组计算和探索性问题。这些练习（大约共100个）的完成需要用现有的软件工具，例如学生或教师自己编写的程序，或数学计算软件包，如MAPLE或Mathematica。不少这种练习为学生提供了通过计算发现新事实或新思想的机会。(有些这类练习在本书的姊妹篇《用MAPLE研究离散数学及其应用》中作了讨论。)

写作题目 每一章后面都有一组应书面完成的题目。学生需要参考数学文献才能做这类题目。有些这类题目涉及历史，可能需要查找原始资料。其他的题目起着联系新题材和新思

想的作用。所有此类练习均向学生展示了正文中没有深入探讨的思想。这些题目把数学概念和书面写作过程结合在一起，以帮助学生探索可供未来研究的领域。（为这些题目准备的推荐文献可以在《学生解题指南》中找到。）

附录　正文有两个附录。第一个介绍指数函数和对数函数，以回顾课程中反复使用的某些基本内容；第二个介绍正文中用以描述算法的伪码。

推荐读物　在正文最后专门有一节为每一章提供推荐读物，其中包括不超过本书水平的书籍、较难的书籍、综述性文章以及发表离散数学新发现的原始文章。

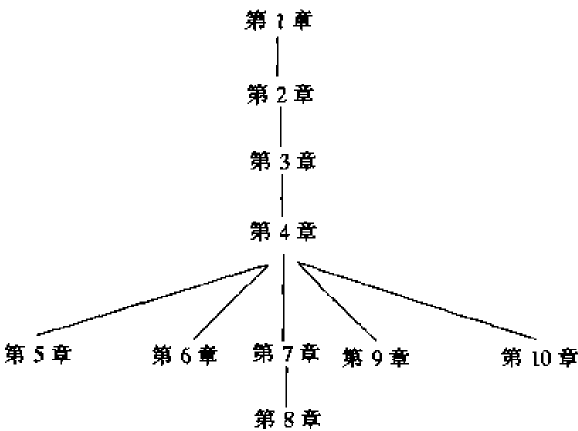
怎样使用本书

本书是精心写作和编排的，适用于不同水平有不同重点的离散数学课程。表1列出了核心章节和供选择的章节。二年级学生离散数学一学期的入门课程可以以核心章节为基础，其他章节由教师取舍。两学期的入门课程可以在核心章节外加上所有可选的数学章节，强调计算机科学的课程可以使用计算机科学的部分或全部可选章节。

表　1

章	核心节	可选的计算机科学章节	可选的数学章节
1	1.1~1.8（按需要定）		
2	2.1~2.3，2.6（按需要定）	2.4	2.5
3	3.1~3.3	3.4，3.5	
4	4.1~4.4	4.7	4.5，4.6
5	5.1，5.5	5.3	5.2，5.4，5.6
6	6.1，6.3，6.5	6.2	6.4，6.6
7	7.1~7.5		7.6~7.8
8	8.1	8.2~8.4	8.5，8.6
9		9.1~9.4	
10		10.1~10.5	

使用本书的教师可以选用或略去每节最后更具挑战性的例题及更加困难的练习，以调整其课程的难度。各章对以前各章的依赖关系如下图所示。



辅助读物

《学生解题指南》(*Student Solution Guide*)　这本可以单独购买的学生手册包括各组练习中用奇数编号的所有习题的完整的解答。这些解答解释了为什么使用某一特定的方法以及为什么这一方法管用。对有些练习还给出了一两种其他可能的解法，用以说明一个问题可以通过不同的途径求解。为每一章后面列出的书面题目推荐的文献可以在本指南找到。本指南包含对学生书写证明题的指导意见，并列出了学生在做离散数学题时常犯的若干错误。本指南还为每一章提供考试样题及解答，以帮助学生准备考试。学生们感到本指南分外有用。

《教师资料手册》(*Instructor's Resource Guide*)　本手册包含所有以偶数编号的练习题的完整解答。它还提出了如何教授每一章教材的建议，包括每一节的重点，以及怎样正确地使用材料。此外，本手册包含每一章的考试样题库，以及样题的解。最后，本手册还给出了样板课程提纲。

《离散数学应用》　这本辅助读物是一本独立的教程，既可结合课本使用，也可独立使用。它包含使用过课本的教师们撰写的 20 多章内容（每章均有自己的练习）。这本书采用与课本类似的格式，各章既可作为一门独立课程的课本，作为学生讨论班的教材，也可供做独立研究的学生使用。这本辅助读物以后再版还计划扩大其覆盖的应用范围。欢迎教师们投稿介绍其他方面的应用，以备再版时采用。

试题库　包含 1300 多个问题的内容广泛的试题库可以在 Windows 系统或 Macintosh 系统下使用。教师可以使用这一软件自己选择或随机产生他们自己的试题。对同样的试题教师可以加上自己的标题和说明，输出同一试题库的不同版本，以及编排已有的问题，或加上自己的问题。在《教师资料手册》中有此题库的打印版，既包括试题也包括解答。

《用 MAPLE 研究离散数学及其应用》　这是一本独立的辅助读物，用以帮助学生使用计算机代数系统 MAPLE 来完成离散数学的各种计算。对本书的每一章，这本新的辅助读物都包括下列内容：对相关的 MAPLE 函数及其使用方法的描述，完成相关计算的 MAPLE 程序，每章结尾处还有阐明怎样使用 MAPLE 作计算和研究的建议和例子，以及可用 MAPLE 做的练习题。

配套网站

已经为本书建立了一个配套网站，该网站将得到不断地维护和改进。网站地址（URL）是http://www.mhhe.com/rosen 。该站点上的网页使你能访问该网站的 5 个不同的部分：

- 关于本书
- 教师资料
- 学生资料
- 离散数学网上指南
- 补充资料

这几部分资料将与本书新版一同问世，当然以后还会增加新材料。

“关于本书”部分包含关于本书及其补充读物的基本信息，还有一个勘误表和一个电子邮件地址，供读者提出勘误和建议。

“教师资料”是网站上的保密部分，它包含有用的工具和资料用以补充课本的不足并提

供离散数学教学经验。

“学生资料”包含对丰富学生学习经验有益的参考文献和补充材料。

“离散数学网上指南”包含带注释的与相关站点的链接，这些站点在书中已用 Web 图标注明（书中凡有 Web 图标之处，网站上均有链接）。本指南中的链接还可用来访问一些站点，这些站点能提供本书中涉及的名人传记、补充材料、最新发现、动画算法和可下载源码。

“补充资料”部分是供师生共用的，包括按章组织的补充教学材料，目的是扩充课本中的内容并使之更清晰。

写给学生

什么是离散数学？离散数学是数学中适用于研究离散对象的那一部分。（这里“离散”的含义是指不同的不连接在一起的元素。）用离散数学能解决的问题包括：

- 有多少种方式可以在一个计算机系统上选择一个合法口令？
- 赢彩票的概率是多少？
- 网络上两台计算机之间是否有通路？
- 使用某一运输系统的两个城市之间的最短路径是什么？
- 怎样把整数列表按增序排列？
- 完成上述排序需要多少步骤？
- 怎样设计两个整数相加的电路？
- 有多少合法的因特网地址？

你们将学习解决上述这类问题需要的离散结构和技术。

更一般地说，在对对象计数时使用离散数学，研究两个有限（或可数）集合之关系时使用离散数学，分析只含有限步数的过程时也使用离散数学。离散数学重要性增长的一个关键因素，是计算机以离散的方式存储和处理信息。

有几条重要的理由需要学习离散数学。首先，通过这一课程你们可以发展自己的数学成熟性，即你理解和创造数学证明的能力。没有这些技巧，你们在数学学科的学习中不可能太深入。

第二、离散数学是通向所有数学学科高级课程的必经之路。离散数学提供许多计算机科学课程的数学基础，这些课程包括数据结构、算法、数据库理论、自动机理论、形式语言、编译理论、计算机安全以及操作系统。没有从离散数学中得到适当的数学基础的学生，在学习上述课程时会遇到更多的困难。有个学生给我发了电子邮件，告诉我她在选修的每一门计算机科学的课程上都用到了本书的知识。

以离散数学为基础的数学课程包括逻辑、集合论、数论、线性代数、抽象代数、组合数学、图论及概率论（其离散部分）。

此外，离散数学还包括解决运筹学（包括许多离散优化技术）、化学、工程和生物学等学科中的问题的必要的数学背景。本书中将学习上述领域的某些应用。

我愿意就如何学好离散数学给同学们提点有益的建议。做练习能使你最大地受益。我建议你尽可能地多做练习，包括正文每一节后的练习和每一章后的补充练习。在参考书后的答案或《学生解题指南》中的答案以前，先要努力自己解题。只有在你已经得出一个解或毫无

头绪时，才查看建议的解答。只有在这种情况下你才能发现《学生解题指南》中对练习的讨论很有帮助。在做练习的时候，不要忘记较难的问题是有标记的。表 2 是难度及相应的记号。

表 2

练习标记	
无标记	常规练习
*	难题
**	特别有挑战性的练习
☞	正文中使用了本练习的某个结论
(需要用到微积分)	解题需要用极限或微积分的概念

最后，我鼓励你们大胆探究超出本书内容的离散数学问题。本书网站的“离散数学网上指南”是个很好的起点，网站地址（URL）是http://www.mhhe.com/rosen。

致　　谢

感谢许多不同学校使用本书的大批师生，是他们向我提出了宝贵的反馈和有益的建议。没有他们的反馈和建议，本书不可能像现在这么好。我要特别感谢 Jerrold Grossman 和 John Michaels 对第 4 版的技术审阅，他们敏锐的“鹰眼”确保了本书的准确性。

我感谢第 1 版、第 2 版、第 3 版以及第 4 版的多位审阅人，他们对我提出了有益的批评并给予鼓励，我希望本版将不辜负他们的厚望。

我还要感谢 McGraw-Hill 公司高等教育组的工作人员对本课题的支持。特别要感谢出版人 J. P. Lenney 的全面支持；感谢倡议此书的编辑 Maggie Rogers 持久的兴趣和热情；感谢开发编辑 Nina Kreiden 的献身和勤勉；感谢编辑助理 Amy Upgren 宝贵的帮助。我同样感谢最早期的编辑 Wayne Yuhasz，是他的洞察力和技巧保证了本书的成功，还要感谢 Jack Shira，Tom Casson 和 Mike Morales，他们协助策划并启动了第 4 版的工作。

对推出第 4 版的工作人员，我要感谢产品主管 Rich DeVitto，设计师 Suzanne Montazer，网站开发者 Ronald Tigges，补充读物协调人 Louis Swaim，以及市场经理 Mary Kittell。

我一直十分感激 AT&T 实验室的管理人员对我的支持。他们为我的专业发展提供了必要的环境，并慷慨地供给我必需的资料，没有这些支持也就没有本书的成功。

Kenneth H. Rosen

作 者 简 介

肯尼斯 H. 罗森(Kenneth H. Rosen)是AT&T荷姆德尔实验室(新泽西州)新概念领域人才部的杰出成员。

罗森博士1972年获密执安大学数学学士学位，1976年获麻省理工学院数学博士学位，其博士论文研究的是数论，导师为H. 斯达克(Harold Stark)。在1982年加入Bell实验室以前，他曾在科罗拉多大学(位于Boulder)和俄亥俄州立大学(位于Columbus)工作，并在缅因大学(位于Orono)担任数学副教授。在AT&T实验室工作时，罗森还在蒙马斯大学计算机科学夜校任教，教授离散数学、编码理论和数据安全的课程。

罗森博士已在专业期刊上发表了数论及数学模型领域的许多论文。他的《初等数论及其应用》一书由Addison-Wesley公司出版，目前已出第3版。由McGraw-Hill公司出版的《离散数学及其应用》已出第4版。这两本教科书均被几百所大学广为采用。他还是《UNIX System V Release 4: An Introduction》的作者之一。此书已售10万册以上，并被译成西班牙文和德文。他也是《Best UNIX Tips Ever》的作者之一，这本书已有中译本。以上两本书均由Osborne McGraw-Hill公司出版。罗森还是《离散数学手册》的编者，这本新书已于1999年由CRC出版社出版。他是CRC离散数学丛书的编辑。罗森的兴趣还包括将数学软件集成为教育环境和专业环境，目前正与Waterloo MAPLE软件公司合作研究这两个方向上的课题。

在Bell实验室以及AT&T实验室期间，罗森博士参与过各种课题的研究，包括运筹学和计算机及数据通信设备生产线设计。他曾协助规划AT&T在多媒体方面的未来产品和服务，包括视频通信、语音识别和图像网络。他还对AT&T使用的新技术作了评估。他发明了许多新式服务，并拥有或申请过多项专利。他开展的更有趣的课题之一涉及在EPCOT中心对AT&T有吸引力的辅助评估技术。

{—° ç Ôs<œ

ì {2! > h Ú t X+h t ÿìUì Y'qŁ \ ÿÔ̧htô! I F'
r Ł¸&ì·ÿp¨Mäh ru ,ÿì pQ'Ip » ÿqÓ{h2%81˛ X
+
§ » öl Ó« ·˛ì ìÿ&ö÷ì O u÷Ñr:hØÒ[vhßO-˛[r:h{òVŒ
I 2ìÞ ßO-ÿ˛[!.3+/ ØÒ[v K [v=I ‹{ {2 2Cÿ{ IÓ¢ ‘ÿÕÞ
ì k ˛Àt!I » ÿqÓ{hì FOL @KJAP IB? R>ÿ"\î ÀI ‡Ò° ÖNâì
Þ» WìFOL @KJAPÀïfflÿ_ w ıý t »
r˛ ÿ ÆH {2 § \t r ! t b <§ h 1ü+$! t b § {Òh
ÿwì'ì=OL JAP * ì ¨y Ü! JAP F AA LDL § h Úh4¶N_ ðI h r
ÿe vy Yu ·ìÿOMHOANRAN ?OO =F=T v ß[> hHtß ìW Öø |I
HtßW"ª I ¸y ˘˘»+r Wr§ ˘˘ 'æ? ì' æq ? Úæ
I ? ì! ')h § Jh I ') Þs ¥ÿxŁ ðÓÓ b r kk‡à! e
Æ Úh« JAPìÞ˘ hÆF AAÿs7 hÆ=LE{2 hØÒ[v % ‡[y4 ì
ÿ¢ı rŁ¸&ì·ÿMä I » ÿqÓ{! ˛À
´¨ JAP F=R=ÿN_?t Øð!|{ ØÒ[vI ° tÕÞI » !f˛ »ÿh Óh
ý¨t ý !¨h'IB? JAPp‹Ù¶Ö…§ kæÞ hfÿÚh° »k Üt h! '
ràkàt! I » Þt bïÿVŒ t b ÿV 'ì JAPÿ Æ9ÀÆØe Ù¶
÷t÷ð ¥ür ð t b § Itâð Ó° N_ Ót bïÿVŒ Ók Üt
h°ýß E÷ðÿht¨¥ N_ï< h°Ó ÜïÿVŒyWð ÀfÿÚ
Ó{ ¾ÿ– wJÓßÿ» hß>÷Ñr: O ìÿtÞhx 2Cÿ{ J˘⅜ xJ
‹SD W`uUÿ÷Ñr: ß>t ÜØÒ[vhxJ ÜI N_'þÿVŒ ï·k » ØÒ[v
ÿ t Ü|{h{òVŒhx b2CE˘ Ó̧ Ó̧ Øh t { °òÿ–
ß> !ìÞ» ÖNâ Wì JAPhF=R=I #by ÀÚh#s+hï· [ÿ',—¸ h wt°
' þh q hIR? KNII A:hÓhÀð t ý § ðh ÚIpÀh ÚìÞ Ó«
‹{ h4¶'2hŒß^RI » ìÞ W I k‡ÖtbEpb
˛ tÓ̧þxæ þ', P ,#2 6+*II',/¨ß¥hk ß> ,ÿ» t ßh ì '
pƒp˘ ÖJhh ! &ht ! Kÿ“ÿ I þ¸ÿ',Û Ł¸F ÿVŒhÓ f
ÿì ',H s Ch ßßÿ«» ïìÞh! I » hÓÕÞe þ', f
Nâ{2E÷òì! ! [v=LE{2h ï· ØFF hì fl· Æ¥ZVŒ ìÞ? !
{2 » h Wì IB? JAPI Às757h wf.ß äŒì 4 {2§Ó hì » 2
4!ÿ2CŒß§Ó> e ß>Óòì4 h 4ì y h tÀ hH >h ^C4hß>òì
y h 4ì4 h ŁÓh§Ó ' ìh“9¸k

目　录

出版者的话
专家指导委员会
译者序
前言
第 1 章　基础：逻辑、集合和函数 …… 1
1.1　逻辑 …… 1
1.1.1　引言 …… 1
1.1.2　命题 …… 1
1.1.3　翻译语言的句子 …… 6
1.1.4　布尔检索 …… 7
1.1.5　逻辑运算和位运算 …… 7
练习 …… 9
1.2　命题等价 …… 15
1.2.1　引言 …… 15
1.2.2　逻辑等价 …… 15
练习 …… 19
1.3　谓词和量词 …… 22
1.3.1　引言 …… 22
1.3.2　量词 …… 23
1.3.3　翻译语句为逻辑表达式 …… 26
1.3.4　选自 Lewis Carroll 的例子（选读） …… 27
1.3.5　绑定变量 …… 28
1.3.6　否定 …… 31
练习 …… 31
1.4　集合 …… 39
1.4.1　引言 …… 39
1.4.2　幂集合 …… 43
1.4.3　笛卡儿积 …… 43
练习 …… 45
1.5　集合运算 …… 47
1.5.1　引言 …… 47
1.5.2　集合相等 …… 49
1.5.3　扩展的并集和交集 …… 51
1.5.4　集合的计算机表示 …… 52
练习 …… 53
1.6　函数 …… 57
1.6.1　引言 …… 57
1.6.2　一对一函数和映上函数 …… 59
1.6.3　反函数和函数组合 …… 61
1.6.4　函数的图像 …… 63
1.6.5　几个重要的函数 …… 64
练习 …… 65
1.7　序列与求和 …… 70
1.7.1　引言 …… 70
1.7.2　序列 …… 70
1.7.3　特殊的整数序列 …… 71
1.7.4　求和 …… 72
1.7.5　基数（选读） …… 75
练习 …… 76
1.8　函数增长 …… 80
1.8.1　引言 …… 80
1.8.2　大 O 符号 …… 80
1.8.3　函数组合的增长 …… 84
1.8.4　大 Ω 和大 Θ 符号 …… 86
练习 …… 88
关键术语和结果 …… 92
复习题 …… 94
补充练习 …… 95
计算机题目 …… 98
计算和研究 …… 98
写作题目 …… 98
第 2 章　基础：算法、整数和矩阵 …… 100
2.1　算法 …… 100
2.1.1　引言 …… 100
2.1.2　搜索算法 …… 102
练习 …… 104
2.2　算法的复杂性 …… 106
2.2.1　引言 …… 106
练习 …… 109
2.3　整数和除法 …… 112
2.3.1　引言 …… 112
2.3.2　除法 …… 112
2.3.3　素数 …… 113

2.3.4 除法算法 …… 115
2.3.5 最大公约数和最小公倍数 …… 115
2.3.6 模运算 …… 117
2.3.7 同余应用 …… 118
2.3.8 密码学 …… 120
练习 …… 121
2.4 整数和算法 …… 124
2.4.1 引言 …… 124
2.4.2 欧几里德算法 …… 125
2.4.3 整数表示 …… 127
2.4.4 整数运算算法 …… 128
练习 …… 131
2.5 数论应用 …… 134
2.5.1 引言 …… 134
2.5.2 若干有用的结果 …… 134
2.5.3 线性同余 …… 136
2.5.4 中国余数定理 …… 137
2.5.5 大整数的计算机算术运算 …… 138
2.5.6 伪素数 …… 140
2.5.7 公钥密码学 …… 141
2.5.8 RSA 加密 …… 141
2.5.9 RSA 解密 …… 142
2.5.10 用 RSA 作公钥系统 …… 143
练习 …… 143
2.6 矩阵 …… 146
2.6.1 引言 …… 146
2.6.2 矩阵运算 …… 147
2.6.3 矩阵乘法运算 …… 148
2.6.4 矩阵的转置和幂 …… 149
2.6.5 0－1 矩阵 …… 150
练习 …… 153
关键术语和结果 …… 156
复习题 …… 158
补充练习 …… 159
计算机题目 …… 161
计算和研究 …… 161
写作题目 …… 162
第 3 章 数学推理 …… 163
3.1 证明方法 …… 163
3.1.1 引言 …… 163
3.1.2 推理规则 …… 164
3.1.3 谬误 …… 167
3.1.4 带量词命题的推理规则 …… 168
3.1.5 证明定理的方法 …… 169
3.1.6 定理与量词 …… 172
3.1.7 停机问题 …… 174
3.1.8 关于证明的一些评注 …… 175
练习 …… 175
3.2 数学归纳法 …… 181
3.2.1 引言 …… 181
3.2.2 良序性 …… 181
3.2.3 数学归纳法 …… 181
3.2.4 数学归纳法证明的例子 …… 183
3.2.5 数学归纳法的第二原理 …… 189
练习 …… 191
3.3 递归定义 …… 195
3.3.1 引言 …… 195
3.3.2 递归地定义函数 …… 196
3.3.3 递归地定义集合 …… 199
练习 …… 201
3.4 递归算法 …… 208
3.4.1 引言 …… 208
3.4.2 递归与迭代 …… 209
练习 …… 211
3.5 程序正确性 …… 212
3.5.1 引言 …… 212
3.5.2 程序验证 …… 213
3.5.3 推理规则 …… 214
3.5.4 条件语句 …… 214
3.5.5 循环不变量 …… 215
练习 …… 217
关键术语和结果 …… 219
复习题 …… 219
补充练习 …… 221
计算机题目 …… 224
计算和研究 …… 225
写作题目 …… 225
第 4 章 计数 …… 227
4.1 计数的基础 …… 227
4.1.1 引言 …… 227
4.1.2 基本的计数原则 …… 227
4.1.3 容斥原理 …… 232
4.1.4 树图 …… 233
练习 …… 234
4.2 鸽巢原理 …… 238
4.2.1 引言 …… 238

4.2.2 推广的鸽巢原理 …… 239
4.2.3 巧妙使用鸽巢原理 …… 240
练习 …… 242
4.3 排列与组合 …… 244
4.3.1 引言 …… 244
4.3.2 排列 …… 244
4.3.3 组合 …… 245
4.3.4 二项式系数 …… 246
4.3.5 二项式定理 …… 248
练习 …… 250
4.4 离散概率 …… 254
4.4.1 引言 …… 254
4.4.2 有限概率 …… 255
4.4.3 事件组合的概率 …… 256
4.4.4 概率的推理 …… 257
练习 …… 258
4.5 概率论 …… 260
4.5.1 引言 …… 260
4.5.2 概率赋值 …… 260
4.5.3 事件的组合 …… 262
4.5.4 条件概率 …… 262
4.5.5 独立性 …… 263
4.5.6 伯努利实验与二项式分布 …… 264
4.5.7 随机变量 …… 266
4.5.8 期望值 …… 267
4.5.9 独立随机变量 …… 269
4.5.10 方差 …… 270
4.5.11 切比雪夫不等式 …… 272
4.5.12 平均状态下的计算复杂性 …… 273
练习 …… 274
4.6 一般性的排列和组合 …… 277
4.6.1 引言 …… 277
4.6.2 有重复的排列 …… 277
4.6.3 有重复的组合 …… 278
4.6.4 具有不可区别物体的集合的排列 …… 281
4.6.5 把物体放入盒子 …… 282
练习 …… 282
4.7 生成排列和组合 …… 286
4.7.1 引言 …… 286
4.7.2 生成排列 …… 286
4.7.3 生成组合 …… 288
练习 …… 289
关键术语和结果 …… 290
复习题 …… 292
补充练习 …… 294
计算机题目 …… 298
计算和研究 …… 298
写作题目 …… 299
第 5 章 高级计数技术 …… 300
5.1 递推关系 …… 300
5.1.1 引言 …… 300
5.1.2 递推关系 …… 300
5.1.3 用递推关系构造模型 …… 301
练习 …… 306
5.2 求解递推关系 …… 312
5.2.1 引言 …… 312
5.2.2 求解常系数线性齐次递推关系 …… 313
5.2.3 常系数线性非齐次的递推关系 …… 317
练习 …… 321
5.3 分而治之关系 …… 325
5.3.1 引言 …… 325
5.3.2 分而治之关系 …… 326
练习 …… 329
5.4 生成函数 …… 330
5.4.1 引言 …… 330
5.4.2 关于幂级数的有用的事实 …… 331
5.4.3 计数问题与生成函数 …… 335
5.4.4 使用生成函数求解递推关系 …… 338
5.4.5 使用生成函数证明恒等式 …… 340
练习 …… 340
5.5 容斥 …… 347
5.5.1 引言 …… 347
5.5.2 容斥原理 …… 347
练习 …… 352
5.6 容斥原理的应用 …… 353
5.6.1 引言 …… 353
5.6.2 容斥原理的另一种形式 …… 354
5.6.3 伊拉脱森筛 …… 355
5.6.4 映上函数的个数 …… 356
5.6.5 错位排列 …… 357
练习 …… 359
关键术语和结果 …… 360
复习题 …… 361
补充练习 …… 362
计算机题目 …… 365

计算和研究 …… 366
写作题目 …… 366
第 6 章 关系 …… 368
6.1 关系及其性质 …… 368
6.1.1 引言 …… 368
6.1.2 函数作为关系 …… 369
6.1.3 集合上的关系 …… 369
6.1.4 关系的性质 …… 370
6.1.5 关系的组合 …… 372
练习 …… 374
6.2 n元关系及其应用 …… 377
6.2.1 引言 …… 377
6.2.2 n 元关系 …… 377
6.2.3 数据库和关系 …… 377
练习 …… 381
6.3 关系的表示 …… 382
6.3.1 引言 …… 382
6.3.2 用矩阵表示关系 …… 382
6.3.3 用图表示关系 …… 384
练习 …… 386
6.4 关系的闭包 …… 387
6.4.1 引言 …… 387
6.4.2 闭包 …… 388
6.4.3 有向图的路径 …… 389
6.4.4 传递闭包 …… 390
6.4.5 沃舍尔算法 …… 393
练习 …… 396
6.5 等价关系 …… 398
6.5.1 引言 …… 398
6.5.2 等价关系 …… 398
6.5.3 等价类 …… 399
6.5.4 等价类与划分 …… 400
练习 …… 402
6.6 偏序 …… 405
6.6.1 引言 …… 405
6.6.2 字典顺序 …… 406
6.6.3 哈斯图 …… 408
6.6.4 极大元素与极小元素 …… 409
6.6.5 格 …… 411
6.6.6 拓扑排序 …… 412
练习 …… 414
关键术语和结果 …… 419
复习题 …… 420
补充练习 …… 422
计算机题目 …… 426
计算和研究 …… 426
写作题目 …… 427
第 7 章 图 …… 428
7.1 图的介绍 …… 428
7.1.1 图的种类 …… 428
7.1.2 图模型 …… 431
练习 …… 432
7.2 图的术语 …… 434
7.2.1 引言 …… 434
7.2.2 基本术语 …… 434
7.2.3 一些特殊的简单图 …… 436
7.2.4 偶图 …… 437
7.2.5 特殊类型的图的一些应用 …… 438
7.2.6 从旧图到新图 …… 440
练习 …… 441
7.3 图的表示和图的同构 …… 443
7.3.1 引言 …… 443
7.3.2 图的表示 …… 443
7.3.3 相邻矩阵 …… 444
7.3.4 关联矩阵 …… 445
7.3.5 图的同构 …… 446
练习 …… 449
7.4 连通性 …… 454
7.4.1 引言 …… 454
7.4.2 通路 …… 454
7.4.3 无向图连通性 …… 455
7.4.4 有向图中的连通性 …… 456
7.4.5 通路与同构 …… 456
7.4.6 统计顶点之间的通路 …… 457
练习 …… 458
7.5 欧拉通路与哈密顿通路 …… 461
7.5.1 引言 …… 461
7.5.2 欧拉回路和欧拉通路的充要条件 …… 462
7.5.3 哈密顿通路和回路 …… 465
练习 …… 468
7.6 最短通路问题 …… 473
7.6.1 引言 …… 473
7.6.2 一个最短通路算法 …… 475
7.6.3 旅行推销员问题 …… 479
练习 …… 480

7.7 平面性图 …… 484
7.7.1 引言 …… 484
7.7.2 欧拉公式 …… 485
7.7.3 库拉图斯基定理 …… 488
练习 …… 489
7.8 图着色 …… 491
7.8.1 引言 …… 491
7.8.2 图着色的应用 …… 495
练习 …… 496
关键术语和结果 …… 499
复习题 …… 501
补充练习 …… 502
计算机题目 …… 507
计算和研究 …… 507
写作题目 …… 508
第 8 章 树 …… 510
8.1 介绍树 …… 510
8.1.1 树作为模型 …… 514
8.1.2 树的性质 …… 516
练习 …… 518
8.2 树的应用 …… 521
8.2.1 引言 …… 521
8.2.2 二叉搜索树 …… 521
8.2.3 决策树 …… 524
8.2.4 前缀码 …… 524
练习 …… 525
8.3 树的遍历 …… 526
8.3.1 引言 …… 526
8.3.2 通用地址系统 …… 527
8.3.3 遍历算法 …… 527
8.3.4 中缀、前缀和后缀记法 …… 533
练习 …… 536
8.4 树与排序 …… 538
8.4.1 引言 …… 538
8.4.2 排序的复杂性 …… 539
8.4.3 冒泡排序 …… 539
8.4.4 归并排序 …… 541
练习 …… 544
8.5 生成树 …… 545
8.5.1 引言 …… 545
8.5.2 一些构造生成树的算法 …… 547
8.5.3 回溯 …… 549
练习 …… 551
8.6 最小生成树 …… 554
8.6.1 引言 …… 554
8.6.2 最小生成树算法 …… 554
练习 …… 558
关键术语和结果 …… 560
复习题 …… 561
补充练习 …… 562
计算机题目 …… 566
计算和研究 …… 566
写作题目 …… 567
第 9 章 布尔代数 …… 568
9.1 布尔函数 …… 568
9.1.1 引言 …… 568
9.1.2 布尔表达式和布尔函数 …… 569
9.1.3 布尔代数中的恒等式 …… 570
9.1.4 对偶性 …… 571
9.1.5 布尔代数的抽象定义 …… 572
练习 …… 573
9.2 布尔函数的表示 …… 574
9.2.1 积之和展开式 …… 574
9.2.2 函数完备性 …… 575
练习 …… 576
9.3 逻辑门电路 …… 577
9.3.1 引言 …… 577
9.3.2 门的组合 …… 578
9.3.3 电路的例子 …… 579
9.3.4 加法器 …… 581
练习 …… 582
9.4 电路的极小化 …… 583
9.4.1 引言 …… 583
9.4.2 卡诺图 …… 584
9.4.3 无需在意条件 …… 588
9.4.4 奎因-莫可拉斯基方法 …… 588
练习 …… 592
关键术语和结果 …… 594
复习题 …… 595
补充练习 …… 596
计算机题目 …… 598
计算和研究 …… 598
写作题目 …… 598
第 10 章 计算模型 …… 600
10.1 语言和文法 …… 600
10.1.1 引言 …… 600

10.1.2 短语结构文法 …… 601
10.1.3 短语结构文法的类型 …… 604
10.1.4 派生树 …… 605
10.1.5 巴科斯-诺尔范式 …… 605
练习 …… 607
10.2 带输出的有限状态机 …… 609
10.2.1 引言 …… 609
10.2.2 带输出的有限状态机 …… 610
练习 …… 614
10.3 不带输出的有限状态机 …… 616
10.3.1 引言 …… 616
10.3.2 串的集合 …… 616
10.3.3 有限状态自动机 …… 617
练习 …… 621
10.4 语言的识别 …… 623
10.4.1 引言 …… 623
10.4.2 正则集合 …… 623
10.4.3 克莱因定理 …… 624
10.4.4 正则集合和正则文法 …… 627
10.4.5 一个不能由有限状态自动机识别的语言 …… 629
10.4.6 一些更强大的机器 …… 630
练习 …… 631
10.5 图灵机 …… 632
10.5.1 引言 …… 632
10.5.2 图灵机的定义 …… 633
10.5.3 用图灵机识别集合 …… 634
10.5.4 用图灵机计算函数 …… 636
10.5.5 不同类型的图灵机 …… 637
10.5.6 丘奇-图灵论题 …… 637
练习 …… 637
关键术语和结果 …… 639
复习题 …… 640
补充练习 …… 641
计算机题目 …… 644
计算和研究 …… 644
写作题目 …… 644
附录 A 指数函数和对数函数 …… 646
附录 B 伪代码 …… 649
奇数练习题答案 …… 654
推荐读物 …… 790
参考文献 …… 794

第1章　基础：逻辑、集合和函数

本章复习离散数学的基础。有三个主要内容：逻辑、集合和函数。逻辑规则给出数学语句的准确含义。例如，逻辑规则告诉我们下列语句的含义："存在一个大于100且是2的幂的整数"，以及"对每个整数 n，小于等于 n 的正整数之和是 $n(n+1)/2$"。逻辑是所有数学推理的基础，对计算机的设计、人工智能、计算机程序设计、程序设计语言以及计算机科学的其他领域，逻辑都有实际的应用。

离散数学的不少内容是研究用于表示离散对象的离散结构的。所有离散结构都从集合构造而来，而集合则指的是一组对象。从集合构造的离散结构例子包括：计数时广为应用的组合，也就是无序的对象集；表示对象之间相互关连的关系，也就是有序偶的集合；图形，也就是顶点集合及连接顶点的边的集合；以及用于模拟计算机的有限状态机。

函数概念是离散数学中特别重要的概念。函数为一个集合中的每个元素恰好指派另一个集合中的某个元素。像序列和字符串等这些有用的结构都是函数的特例。函数还用来表示一个程序过程解题的步数。分析算法时使用的术语和概念与函数增长有关。递归函数指定用较小正整数时的函数值来定义较大正整数时的函数值，递归函数用于解决许多计数问题。

1.1　逻辑

1.1.1　引言

逻辑规则给出数学语句的准确含义，这些规则用来区分有效和无效的数学论证。由于本书的一个主要目的是教会读者如何理解和如何构造正确的数学论证，所以我们从介绍逻辑开始离散数学的学习。

逻辑不仅对理解数学推理十分重要，而且在计算机科学中有许多应用。这些逻辑规则用于计算机电路设计、计算机程序构造、程序正确性证明以及许多其他方面。在随后的几章中将逐一讨论这些应用。

1.1.2　命题

我们从介绍逻辑的基本成分——命题开始我们的讨论。一个命题是一个或真或假的语句，但不能既真又假。

例1　下述语句均为命题。

1. 华盛顿是美国的首都。
2. 多伦多是加拿大的首都。
3. $1+1=2$。
4. $2+2=3$。

命题1和3成真，命题2和4为假。■

下一例子给出了不是命题的若干语句。

例 2 考虑下述语句：

1. 几点了？
2. 仔细读这个。
3. $x+1=2$。
4. $x+y=z$。

语句 1 和 2 不是命题，因为它们不是陈述语句。语句 3 和 4 不是命题，因为它们既不成真，也不为假，这是由于语句中的变量没有被赋值。在 1.3 节将讨论这一类语句形成命题的多种方法。■

命题常用字母来表示，就像用字母表示变量那样。习惯上用来表示命题的字母是 p、q、r、s... 等。如果一个命题是真命题，它的真值为真，用 T 表示；如果它是假命题，其真值为假，用 F 表示。

现在我们转而注意从那些已有的命题产生新命题的方法，这些方法在 1854 年曾由英国数学家布尔[⊖]在他的题为《*The Laws of Thought*》的书中讨论过。许多数学陈述都是组合一个或多个命题而来。称为**复合命题**的新命题由已有的命题用逻辑运算符组合而来。

定义 1 令 p 为一命题，则语句

"不是 p 所说的情形。"

是另一个命题，称为 p 的**否定**。p 的否定用 $\neg p$ 表示。命题 $\neg p$ 读作"非 p"。

例 3 找出命题

"今天是星期五。"

的否定，并用中文表示。

解： 否定为

"并非今天是星期五。"

也可以更简单地表达为

"今天不是星期五。" ■

注意 严格地说，像例 3 这种含有可变时间的语句不是命题，除非假定了一个确定

⊖ 乔治·布尔（George Boole，1815—1864）是皮匠的儿子，1815 年 11 月生于英格兰的林肯。由于家境贫寒，布尔不得不在协助养家的同时为自己能受教育而奋斗。不管怎么说，他成了 19 世纪最重要的数学家之一。尽管他考虑过以牧师为业，但最终还是决定从教，并且不久就开办了自己的学校。在备课的时候，布尔不满意当时的数学课本，便决定阅读伟大数学家的论文。在阅读伟大的法国数学家拉格朗日的论文时，布尔有了变分方面的新发现。变分是数学分析的一个分支，它处理的是寻求优化某些参数的曲线和曲面。

1848 年，布尔出版了《*The Mathematical Analysis of Logic*》，这是他对符号逻辑诸多贡献中的第一次。1849 年，他被任命为位于爱尔兰科克的皇后学院的数学教授。1854 年，他出版了《*The Laws of Thought*》，这是他最著名的著作。在这本书中布尔介绍了现在以他的名字命名的布尔代数。布尔撰写了微分方程和差分方程的课本，这些课本在英国一直使用到 19 世纪末。布尔在 1855 年结婚，他的妻子是皇后学院一位希腊文教授的侄女。1864 年，布尔死于肺炎，肺炎是他在暴风雨天气中尽管已经湿淋淋的了仍坚持上课引起的。

的时间。同样，除非假定了确定的地点，否则含有可变地点的语句不是命题；除非假定了确定的人，否则含有可变代词的语句不是命题。

真值表给出命题真值之间的关系。在确定由较简单命题组成的命题之真值时，真值表特别有用。表 1-1 给出的是命题及其否定所有可能的真值。

命题的否定也可以看作非运算符作用在命题上的结果。非运算符从一个已有的命题构造出一个新命题。现在将引入从两个或多个已有命题构造新命题的逻辑运算符，这些逻辑运算符也称为联接词。

表 1-1 命题之否定的真值

p	$\neg p$
T	F
F	T

定义 2 令 p 和 q 为命题。用 $p\wedge q$ 表示的命题“p 而且 q”是这样一个命题：当 p 和 q 均成真时它成真，否则为假。命题 $p\wedge q$ 称为 p 和 q 的合取。

表 1-2 给出了 $p\wedge q$ 的真值表。注意真值表中共有四行，每行对应着命题 p 和 q 真值的一种组合。

例 4 找出命题 p 和 q 的合取，其中 p 为命题“今天是星期五”，q 为命题“今天下雨”。

解 这两个命题的合取 $p\wedge q$ 是命题“今天是星期五而且下雨”。这一命题在下雨的星期五成真，不是星期五的日子为假，不下雨的星期五也为假。■

定义 3 令 p 和 q 为命题，用 $p\vee q$ 表示的命题“p 或 q”是这样一个命题：它的真值在 p 和 q 均为假时为假，否则成真。命题 $p\vee q$ 称为 p 和 q 的析取。

$p\vee q$ 的真值表如表 1-3 所示。联结词“或”在析取中的使用对应于词“或”（or）包含的两种情况之一，即是“同或”（inclusive or）。析取所含两命题之一成真或两者均成真时，析取的真值为真。例如，下面这句话中，“或”即是以“同或”的方式使用的：

“选修过微积分或计算机科学的学生可以选修本课。”

这里我们指的是，选修过微积分和计算机科学两门课的学生以及只选修过其中一门课的学生都可以选修本课。另一方面，当我们说：

“学过微积分或学过计算机科学，但不是两者都学过的学生，可以注册本课。”的时候，使用的“或”是“异或”。这里我们的意思是既学过微积分，又学过计算机科学的学生不能选修本课；只有那些恰好在这两门课中选修过一门的学生可以选修本课。

表 1-2 两命题合取的真值表

p	q	$p\wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

表 1-3 两命题析取的真值表

p	q	$p\vee q$
T	T	T
T	F	T
F	T	T
F	F	F

同样，若餐馆的菜单上写着“汤或沙拉，加一道主菜。”一般这都表示顾客可以喝汤，也可以吃沙拉，但不是既有汤也有沙拉。因此，这里是“异或”而不是“同或”。

例 5 若 p 和 q 就是例 4 中的两个命题，它们的析取是什么?

解 p 和 q 的析取 $p \vee q$ 是命题

“今天是星期五，或今天下雨。”

这一命题在星期五或下雨天（包括下雨的星期五）的任何一天都成真。只有在既不是星期五，又不下雨的日子，此命题为假。 ■

前面已经说过，在析取中使用的联结词“或”(or) 对应于“或”包含的两种情形之一，即“同或”。所以当析取中的两个命题之一成真或两者均成真时，析取成真。有时我们也按“异或”的含义使用“或”。用异或来联结命题 p 和 q 时，就得到命题“p 或 q（但非两者)”。这一命题当 p 成真且 q 为假时成真，或反过来当 p 为假且 q 成真时也成真。p 和 q 两者均为假或均成真时，这一命题为假。

定义 4 令 p 和 q 为命题。p 和 q 的异或，用 $p \oplus q$ 表示，是这样一个命题：当 p 和 q 中恰有一个成真时它成真，否则它为假。

两个命题异或的真值表如表 1-4 所示。我们还将讨论其他几个重要的命题组合方式。

定义 5 令 p 和 q 为命题。蕴含 $p \to q$ 是这样一个命题：在 p 成真而 q 为假时它为假，否则成真。在这一蕴含命题中，p 称为假设（或前项，前提），q 称为结论（或后果）。

蕴含命题 $p \to q$ 的真值表如表 1-5 所示。

表 1-4 两命题异或的真值表

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

表 1-5 蕴含 $p \to q$ 的真值表

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

由于数学推理中许多地方出现蕴含，表示 $p \to q$ 的术语很多，下面是常用的几个：

- “如果 p，那么 q”
- “p 蕴含 q”
- “如果 p，q”
- “p 仅当 q”
- “p 是 q 的充分条件”
- “q 如果 p”
- “q 每当 p”
- “q 是 p 的必要条件”

注意，只有在 p 成真而 q 为假时 $p \to q$ 才为假，所以当 p 和 q 均成真，或 p 为假（无论 q 的真值是什么）时，其真值为真。

有助于记住“当蕴含命题的前提为假时其值为真”这一事实的方法是把蕴含想像为合同或义务。如果这一陈述中规定的条件不成立，也就没有义务。

例如，“如果你挣的钱超过 \$25 000，那你必须备案交税”这一语句与挣钱少于

$25 000的人无关。如果你挣的钱超过 $25 000 而不备案交税，你没尽义务。同样，在垒球手的合同中写下“若垒球手击出 60 个以上的本垒打，奖金 1 000 万美元”，则只有当球手击出 60 多个本垒打而没有得到上述奖金时，才是违反合同。当球手的本垒打少于 60 次时，上述合同不起任何效用。

我们定义的蕴含比语言中的蕴含含义更广泛一些。例如，在

“如果今日天晴，那么我们将去海滩。”

中有假设和结论之间的联系，这是一般语言中的蕴含。而且除非今日的确天晴但我们不去海滩，否则上述蕴含总是成立。另一方面，根据蕴含命题的定义，蕴含语句

“如果今天是星期五，那么 $2+3=5$。”

总是成立的，因为它的结论是成真的（于是假设部分的真值不起作用）。蕴含语句

“如果今天是星期五，那么 $2+3=6$。”

是除星期五以外天天成真，尽管 $2+3=6$ 为假。

在日常对话时，我们不会使用最后这两个蕴含命题，因为其中的假设和结论之间没有什么联系。在数学推理中我们考虑的蕴含命题比语言中使用的要广泛一些。蕴含作为一个数学概念不依赖于假设和结论之间的因果关系。我们关于蕴含的定义规定了它的真值，而这一定义不是以语言的用法为基础的。

许多程序设计语言中使用的 if－then（如果－那么）结构与逻辑中使用的不同。大部分程序设计语言中都有 **if** p **then** S 这样的语句，其中 p 是命题而 S 是一个程序段（待执行的一条或多条语句）。当程序的运行遇到这样一条语句时，如果 p 为真，就执行 S；但若 p 为假，则 S 不执行。下面的例子说明了这一点。

例 6 若执行语句

if $2+2=4$ **then** $x := x+1$

之前，$x=0$，执行以后 x 的值是什么？（符号:=代表赋值，语句 $x := x+1$ 表示将 $x+1$ 的值赋给 x。）

解 因为 $2+2=4$ 为真，赋值语句 $x := x+1$ 被执行。因此在执行此语句之后 x 的值是 $0+1=1$。 ■

我们可以用非运算符和已定义的各联结词构造复合命题。小括号用于规定复合命题中多个逻辑运算符的操作顺序，最内层小括号里面的逻辑运算符首先操作。例如，$(p \vee q) \wedge (\neg r)$ 是 $p \vee q$ 和 $\neg r$ 的合取。为减少所需的小括号数目，我们规定非运算符先于其他逻辑运算符操作。这表示 $\neg p \wedge q$ 是 $\neg p$ 和 q 的合取，亦即 $(\neg p) \wedge q$，而不是 p 与 q 的合取非，即不是 $\neg(p \wedge q)$。

由 $p \rightarrow q$ 可以构成几个相关的蕴含。命题 $q \rightarrow p$ 称为 $p \rightarrow q$ 的**逆蕴含**，而 $p \rightarrow q$ 的**倒置命题**是命题 $\neg q \rightarrow \neg p$。

例 7 找出蕴含命题

“如果今天是星期四，那么我今天有考试。”

的逆命题和倒置命题。

解 逆命题是

"如果我今天有考试，那么今天是星期四。"

而倒置命题是

"如果我今天没有考试，那么今天不是星期四。" ■

现在我们引入另一种组合命题的方式。

定义 6 令 p 和 q 为命题。双蕴含 $p \leftrightarrow q$ 是这样一个命题：其真值只有在 p 和 q 有同样的真值时为真，否则为假。

$p \leftrightarrow q$ 的真值表如表 1-6 所示。注意，双蕴含 $p \leftrightarrow q$ 恰在 $p \to q$ 和 $q \to p$ 两个蕴含均为真时为真。正因为如此，术语

"p 当且仅当 q"

常用来表示这一双蕴含。表达命题 $p \leftrightarrow q$ 的其他方式是："p 是 q 的充分必要条件" 以及 "如果 p，那么 q；反之亦然。"

表 1-6 双蕴含 $p \leftrightarrow q$ 的真值表

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

1.1.3 翻译语言的句子

有许多理由需把语言的句子翻译成由命题变量和逻辑联结词组成的表达式，特别是因为语言（包括一切人类语言）常有二义性，把句子译成逻辑表达式可以消除歧义。注意，做这种翻译也许要在句子含义的基础上做些合理的假设。此外，一旦我们完成了从句子到逻辑表达式的翻译，我们就可以分析这些逻辑表达式以决定它们的真值。我们还可以对它们进行处理，并用推理规则（参见第 3 章）对它们做推理分析。

为说明把语句翻译成逻辑表达式的过程，考虑下面的例子。

例 8 怎样把下面的句子翻译成逻辑表达式？

"只有你主修计算机科学或不是新生，才可以从校园内访问因特网。"

解 有许多方法翻译这一句子为逻辑表达式。尽管可以用一个命题变量，例如 p 来表示这一句子，但在分析其含义或用它做推理时，这种表示不会有什么作用。我们的办法是用命题变量表示其中的每一个句子成分，并找出其间合适的逻辑联结词。具体地说，令 a、c 和 f 分别表示 "你可以从校园内访问因特网"、"你主修计算机科学" 和 "你是个新生"。注意到 "只有... 才" 是表达蕴含的一种方式，上述句子可以译为

$$a \to (c \vee \neg f)$$ ■

例 9 怎样把下面的句子翻译成逻辑表达式？

"除非你已满 16 周岁，否则只要你身高不足 4 英尺[⊖]就不能乘公园滑行铁道。"

解 有许多方式把这一句子翻译成逻辑表达式。最简单也最无用的方式是用一个命题变量，例如 p 表示这一句子。尽管这样做并不错，但当我们尝试分析这一句子，或用它做推理时，这种翻译对我们不会有什么帮助。较合适的方法是用命题变量表示其中的每一个句子成分，再找出它们之间合适的逻辑联结词。具体地说，令 p、r 和 s 分别表示"你能乘公园滑行铁道"、"你身高不足 4 英尺"和"你已满 16 周岁"，则上述句子可以翻译为

$$(r \wedge \neg s) \rightarrow \neg q$$

当然，还有其他方式可以把上述句子表示为逻辑表达式，但上面我们使用的这一表达式已满足我们的需要。 ■

1.1.4 布尔检索

逻辑联结词广泛用于在大量信息中检索，例如，检索网页索引。由于这些检索使用来自命题逻辑的技术，所以称为*布尔检索*。

在布尔检索中，联结词 *AND* 由于匹配包含两个检索项的记录，联结词 *OR* 用于匹配两个检索项之一或两项均匹配的记录，而联结词 *NOT*（有时写作 *AND NOT*）用于排除某个特定的检索项。当用布尔检索为有潜在价值的信息定位时，常需要细心安排逻辑联结词的使用。下面的例子说明布尔检索是怎样执行的。

例 10 网页检索的布尔检索技术。大部分网上检索引擎支持布尔检索，因为它有助于找到有关特定主题的网页。例如，要用布尔检索找出关于新墨西哥州（New Mexico）各大学的网页，我们可以寻找与 NEW *AND* MEXICO *AND* UNIVERSITIES 匹配的网页，检索的结果将包括含 NEW（新）、MEXICO（墨西哥）和 UNIVERSITIES（大学）三个词的那些网页。这里面包含了所有我们感兴趣的网页，还包括了有关墨西哥国的新大学的网页。另一例子是，要找出与新墨西哥州或亚利桑那州（Arizona）的大学有关的网页，我们可以检索与（NEW *AND* MEXICO *OR* ARIZONA）*AND* UNIVERSITIES 匹配的网页。（注意，其中联结词 *AND* 优先于联结词 *OR*。）这一检索的结果将包括含 UNIVERSITIES 一词和 NEW 与 MEXICO 两词的所有网页及含 UNIVERSITIES 一词和 ARIZONA 一词的所有网页。同样除这两类我们感兴趣的网页外还会列出其他网页。最后，要想找出有关墨西哥国（不是新墨西哥州）的大学的网页，我们可以先找与 MEXICO *AND* UNIVERSITIES 匹配的网页，但由于这一检索的结果将会包括有关新墨西哥州的大学的网页以及墨西哥国的大学的网页，所以更好的办法是检索与（MEXICO *AND* UNIVERSITIES）*NOT* NEW 匹配的网页。这一检索的结果将包括含 MEXICO 和 UNIVERSITIES 两个词但不含词 NEW 的所有网页。 ■

1.1.5 逻辑运算和位运算

计算机用字位表示信息，每个字位有两个可能的值，即 0 或 1。字位的这一含义来自

㊀ 1 英尺 = 30.48 厘米。——译者注

二进制数字，因为 0 和 1 是用在数的二进制表示中的数字[1]。1946 年著名的统计学家图凯(John TuKey)[2]引入了这一术语。字位可以用于表示真值，因为只有两个真值，即真与假。习惯上用 1 表示真，用 0 表示假。因此，1 表示 T（真），0 表示 F（假）。若变量的值或为真或为假，则此变量称为布尔变量。于是一个布尔变量可以用一个字位表示。

计算机的字位运算对应于逻辑联结词，只要在运算符∧、∨和⊕的真值表中用 1 代替 T，用 0 代替 F，就能得到表 1-7 所示的对应的字位运算表。我们还将用符号 *OR*、*AND* 和 *XOR* 表示运算符∨、∧和⊕，许多程序设计语言正是这样表示的。

信息一般用位串表示，也就是用 0 和 1 的序列表示。这时，对位串的运算即可用来处理信息。

表 1-7　字位运算符 *OR*、*AND* 和 *XOR* 的真值表

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

定义 7　位串是 0 个或多个字位的序列。位串的长度就是它所含的字位的个数。

例 11　101010011 是长度为 9 的一个位串。■

我们可以把字位运算扩展到位串上。我们在长度相同的两个位串上定义它们的 bitwise *OR*（按位 *OR*）、bitwise *AND*（按位 *AND*）和 bitwise *XOR*（按位 *XOR*）分别为这样的位串，其中每个字位均由被运算的两个位串对应字位经 *OR*、*AND* 和 *XOR* 运算而得。我们分别用符号∨、∧和⊕表示按位 *OR*、按位 *AND* 和按位 *XOR* 等运算。我们用下面的例子解释对位串的这些按位运算。

---

[1] 历史注记：有人建议过用别的词称呼二进制数字，例如，binit 和 bigit，但从来没有被广泛采用过。可能是因为 bit 作为英语常用词的含义较合适，因而它被采用了。要了解图凯选用 bit 一词的报道，参见 1984 年 4 月期的《*Annals of the History of Computing*》。

[2] 图凯(John Wilder Tukey)1915 年生于麻省的 New Bedford，是个独生子。他的双亲都是教员，认为家庭教育最适合开发他的潜力。他的正规教育从布朗大学开始，学的是数学和化学。他在布朗大学获得化学硕士学位，接着在普林斯顿大学继续学习，不过从化学改学数学。1939 年由于他在拓扑学方面的工作，获得普林斯顿大学博士学位，同时还被任命为普林斯顿大学数学教师。随着第二次世界大战的爆发，他参加了火力控制研究办公室(Fire Contorl Research Office)的工作，开始研究统计学。图凯发现统计研究很合他的口味，并以他的技能给几位最有影响的统计学家留下了深刻印象。1945 年二战结束时，图凯回到普林斯顿大学数学系担任教授，并在 AT&T 贝尔实验室兼职。图凯于 1966 年创立了普林斯顿大学统计系，成为该系首任主任。图凯在统计学的许多领域作出了重要贡献，包括方差分析、时间序列谱评估、关于采自一台设备的一组参数值的推断以及统计原理。不过他最著名的工作是他与库雷（J. W. Cooley）共同发明的快速傅里叶变换。

服务于总统科学顾问委员会期间，图凯以他的洞察力和经验作出了重要贡献。他曾担任过处理环境、教育以及化学与健康等事务的好几个重要委员会的主席。他还参与了几个从事核裁军工作的委员会。图凯得过许多奖，包括国家科学奖章。

例 12 求位串 01 1011 0110 和 11 0001 1101 的按位 *OR*、按位 *AND* 和按位 *XOR*（这里以及本书其他地方均把位串按 4 个字位分块以便于阅读）。

解 这两个位串的按位 *OR*、按位 *AND* 和按位 *XOR* 分别由对应字位的 *OR*、*AND* 和 *XOR* 得到，其结果是

```
01 1011 0110
11 0001 1101
------------
11 1011 1111    按位 OR
01 0001 0100    按位 AND
10 1010 1011    按位 XOR
```

■

练习

1. 下列哪些语句是命题？这些是命题的语句的真值是什么？
   a）波士顿是马萨诸塞州首府。
   b）迈阿密是佛罗里达州首府。
   c）$2+3=5$。
   d）$5+7=10$。
   e）$x+2=11$。
   f）回答这一问题。
   g）对每一对实数 x、y，都有 $x+y=y+x$。
2. 下列哪些是命题？这些命题的真值是什么？
   a）别过来。
   b）几点了？
   c）在缅因州没有黑苍蝇。
   d）$4+x=5$。
   e）若 $x=1$，则 $x+1=5$。
   f）若 $x=z$，则 $x+y=y+z$。
3. 下列各命题的否定是什么？
   a）今天是星期四。
   b）新泽西没有污染。
   c）$2+1=3$。
   d）缅因州的夏天又热又晒。
4. 令 p、q 为如下命题：
   p：本周我买了一张彩票。
   q：星期五我中了百万元头奖。
   把下列各命题表达为汉语句子：
   a）$\neg p$　　b）$p \vee q$
   c）$p \rightarrow q$　　d）$p \wedge q$
   e）$p \leftrightarrow q$　　f）$\neg p \rightarrow \neg q$
   g）$\neg p \wedge \neg q$　　h）$\neg p \vee (p \wedge q)$

5. 令 p、q 为如下命题：
   p：气温在零度以下。
   q：正在下雪。
   用 p、q 和逻辑联结符写出下列各命题：
   a) 气温在零度以下且正在下雪。
   b) 气温在零度以下，但不在下雪。
   c) 气温不在零度以下，也不下雪。
   d) 也许在下雪，也许在零度以下（也许两者在内）。
   e) 若气温在零度以下，那也就在下雪。
   f) 也许气温在零度以下，也许在下雪，但如果在零度以下，就不在下雪。
   g) 气温在零度以下是下雪的充分必要条件。
6. 令 p、q 和 r 为如下命题：
   p：你得流感了。
   q：你错过了最后的考试。
   r：这门课你通过了。
   将下列各命题用汉语表示：
   a) $p \to q$
   b) $\neg q \leftrightarrow r$
   c) $q \to \neg r$
   d) $p \vee q \vee r$
   e) $(p \to \neg r) \vee (q \to \neg r)$
   f) $(p \wedge q) \vee (\neg q \wedge r)$
7. 令 p、q 为如下命题：
   p：你的车速超过每小时 65 英里（1 英里 = 1.6 公里）。
   q：你接到一张超速罚单。
   用 p、q 和逻辑联结符写出下列命题：
   a) 你的车速没有超过每小时 65 英里。
   b) 你的车速超过每小时 65 英里，但没接到超速罚单。
   c) 你的车速若超过每小时 65 英里，将接到一张超速罚单。
   d) 你的车速不超过每小时 65 英里，就不会接到超速罚单。
   e) 车速超过每小时 65 英里足以接到超速罚单。
   f) 你接到一张超速罚单，但你的车速没超过每小时 65 英里。
   g) 只要你接到一张超速罚单，你的车速就超过每小时 65 英里。
8. 令 p、q、r 为如下命题：
   p：你的期末考试得了个 A。
   q：你做了本书每一道练习。
   r：这门课你得了个 A。
   用 p、q、r 和逻辑联结符写出下列命题：
   a) 这门课你得了个 A，但你并没做本书的每道练习。

b）你的期末考试得了个 A，你做了本书的每一道练习，并且这门课你得了个 A。
c）想在这门课得 A，你必须在期末考试得 A。
d）你的期末考试得了个 A，你没有做本书的每道练习，可不管怎样这门课你得了个 A。
e）期末考试得 A 并且做本书的每道练习，足以使你这门课得 A。
f）这门课得 A 当且仅当你做本书的每道练习或期末考试得 A。

9. 判断下列各蕴含是真是假：
a）若 1+1=2，则 2+2=5。
b）若 1+1=3，则 2+2=4。
c）若 1+1=3，则 2+2=5。
d）若猪会飞，那么 1+1=3。
e）若 1+1=3，就存在上帝。
f）若 1+1=3，猪就会飞。
g）若 1+1=2，猪就会飞。
h）若 2+2=4，则 1+2=3。

10. 就下列各语句，判断其中想表达的是同或还是异或，说明理由：
a）要求有使用 C++ 或 Java 的经验。
b）午餐包括汤或沙拉。
c）你必须持护照或选民登记卡才能入境。
d）出版或销毁。

11. 对下列各语句，说一说其中的或是同或（即析取）与异或时它们的含义。你认为语句想表示的是哪个或？
a）要选修离散数学课，你必须已经选修微积分或计算机科学的一门课。
b）从 Acme 汽车公司购买一部新车，你就能得到 2 000 美元现金回扣，或 2% 的汽车贷款。
c）两人套餐包括 A 列中的两项或 B 列中的三项。
d）若下雪超过两英尺或寒流低于 −100，学校就停课。

12. 在古西西里的传说中有一个住在边远小镇上的剃头匠，只有穿过一条危险的山路才能找到他。这个剃头匠只给那些不自己剃须的人刮胡子。这样的剃头匠能存在吗？

13. 边远村庄的每个人要么总说真话，要么总说谎。对旅游者的问题，村民要么回答“是”，要么回答“不”。假定你在这一地区旅游，走到了一个岔路口，一条岔路通向你想去的遗址，另一岔路通向丛林深处。此时恰有一村民站在岔路口，问村民什么样的一个问题就能决定走哪条路？

14. 一个探险者被几个吃人者抓住了。有两种吃人者：总是说谎的和永不说谎的。除非探险者能判断出一位指定的吃人者是说谎者还是说真话者，否则就要被吃人者烤了吃。探险者只被允许问这位吃人者一个问题。
a）解释为什么问：“你说谎吗？”是不行的。
b）找一个问题，使探险者可以用来判断该吃人者是说谎者还是说真话者。

15. 把下列语句写成“如果 p，那么 q”的形式。［提示：参考本节列出的通常表达蕴含的方式。］
a）吹东北风的时候就下雪。

b）暖天持续一周苹果树就开花。
c）活塞队赢得冠军就意味着他们打败了湖人队。
d）必须走 8 英里才能到朗斯峰的顶峰。
e）要得到教授职位，世界闻名就够了。
f）如果你驾车超过 400 英里，就需要买汽油了。
g）只有你购买的 CD 机不超过 90 天，你的保修单才有效。

16. 把下列语句写成“如果 p，那么 q”的形式。[提示：参考本节列出的通常表达蕴含的方式。]
a）只要你发给我一个电子邮件，我就会记住把地址寄给你。
b）要成为美国公民，只要你生在美国就行了。
c）如果你保存课本，它会是你未来其他课程的有用参考书。
d）如果守门员表现好，红翼队将赢得斯坦利杯。
e）你获得这一职位表明你有最好的信誉。
f）有风暴时沙滩受侵蚀。
g）在服务器登录必须有一个有效的口令。

17. 把下列命题写成“p 当且仅当 q”的形式：
a）如果外边热你就买冰激淋；如果你买冰激淋，外边就热。
b）你赢得竞赛的充分必要条件是你有那惟一的获胜卷。
c）只有你有关系才能得到提拔，只有得到提拔你才有关系。
d）如果你看电视，记忆会衰退；反之亦然。
e）火车恰在我乘坐的那些日子晚点。

18. 把下列命题写成“p 当且仅当 q”的形式：
a）你能在这门课得 A 的充分必要条件是学会解离散数学问题。
b）如果你每天看报，你就了解情况；反之亦然。
c）周末天下雨，下雨天是周末。
d）仅当巫师不在家时你能看到他，仅当你能看到巫师时他不在家。

19. 给出下列各蕴含关系的逆和倒置：
a）如果今天下雪，我明天就去滑雪。
b）只要有测验，我就来上课。
c）只有当正整数没有 1 和它自己以外的因数时，它才是素数。

20. 给出下列蕴含关系的逆和倒置：
a）如果今晚下雪，我将呆在家里。
b）只要是阳光充足的夏天，我就去海滩。
c）如果我起床晚了，一定是我白天睡到了中午。

21. 为下列各复合命题构造真值表：
a）$p \wedge \neg p$
b）$p \vee \neg p$
c）$(p \vee \neg q) \rightarrow q$
d）$(p \vee q) \rightarrow (p \wedge q)$

e) $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$

f) $(p \rightarrow q) \leftrightarrow (q \rightarrow p)$

22. 为下列各复合命题构造真值表：

a) $p \oplus p$

b) $p \oplus \neg p$

c) $p \oplus \neg q$

d) $\neg p \oplus \neg q$

e) $(p \oplus q) \vee (p \oplus \neg q)$

f) $(p \oplus q) \wedge (p \oplus \neg q)$

23. 为下列各复合命题构造真值表：

a) $p \rightarrow \neg q$

b) $\neg p \leftrightarrow q$

c) $(p \rightarrow q) \vee (\neg p \rightarrow q)$

d) $(p \rightarrow q) \wedge (\neg p \rightarrow q)$

e) $(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$

f) $(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$

24. 为下列各复合命题构造真值表：

a) $(p \vee q) \vee r$

b) $(p \vee q) \wedge r$

c) $(p \wedge q) \vee r$

d) $(p \vee q) \wedge r$

e) $(p \vee q) \wedge \neg r$

f) $(p \wedge q) \vee \neg r$

25. 为下列各复合命题构造真值表：

a) $p \rightarrow (\neg q \vee r)$

b) $\neg q \rightarrow (q \rightarrow r)$

c) $(p \rightarrow q) \vee (\neg p \rightarrow r)$

d) $(p \rightarrow q) \wedge (\neg p \rightarrow r)$

e) $(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$

f) $(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$

26. 构造$((p \rightarrow q) \rightarrow r) \rightarrow s$ 的真值表。

27. 构造$(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow s)$的真值表。

28. 假定在计算机程序中，执行下列语句之前，$x = 1$，在执行之后 x 的值是什么？

a) **if** $1 + 2 = 3$ **then** $x := x + 1$

b) **if** $(1 + 1 = 3)$ *OR* $(2 + 2 = 3)$ **then** $x := x + 1$

c) **if** $(2 + 3 = 5)$ *AND* $(3 + 4 = 7)$ **then** $x := x + 1$

d) **if** $(1 + 1 = 2)$ *XOR* $(1 + 2 = 3)$ **then** $x := x + 1$

e) **if** $x < 2$ then $x := x + 1$

29. 求下列各对位串的按位 *OR*、按位 *AND* 及按位 *XOR*：

a) 101 1110，010 0001

b) 1111 0000，1010 1010

c) 00 0111 0001，10 0100 1000

d) 11 1111 1111，00 0000 0000

30. 计算下列表达式：

a) 1 1000 ∧ (0 1011 ∨ 1 1011)

b) (0 1111 ∧ 1 0101) ∨ 0 1000

c) (0 1010 ⊕ 1 1011) ⊕ 0 1000

d) (1 1011 ∨ 0 1010) ∧ (1 0001 ∨ 1 1011)

模糊逻辑可用于人工智能。在模糊逻辑中命题的真值是界于 0 和 1（包括 0 和 1）之间的数。以 0 为真值的命题为假，以 1 为真值的命题为真。0 和 1 之间的真值表示不同程度的真值。例如，语句“傅雷德是幸福的”的真值可以是 0.8，因为傅雷德大部分时间是幸福的；“约翰是幸福的”的真值可能是 0.4，因为他幸福的时间比一半稍短。

31. 模糊逻辑中命题否定的真值是 1 减去该命题的真值。语句“傅雷德不幸福”和“约翰不幸福”的真值是什么？

32. 模糊逻辑中两个命题的合取的真值是两个命题真值的最小值。语句“傅雷德和约翰都幸福”和“傅雷德和约翰都不幸福”的真值是什么？

33. 模糊逻辑中两个命题的析取的真值是两个命题真值的最大值。语句“傅雷德幸福或约翰幸福”与“傅雷德不幸福或约翰不幸福”的真值是什么？

*34. 断言“本语句为假”是命题吗？

如果能给一组命题表达式中的每个变量一个真值，使各表达式均为真，则这一组命题表达式是一致的。在给出系统规范时，必须使这些规范一致。

35. 下列规范一致吗？

“当且仅当系统正常操作时，系统处于多用户状态。如果系统正常操作，则它的核心程序正在运行。核心程序不能正常运行，或者系统处于中断模式。如果系统不处于多用户状态，它就处于中断模式。系统不处在中断模式。”

36. 下列规范一致吗？

“如果文件系统未加锁，那么新消息将被排成队。如果文件系统未加锁，则系统正常运行；反之亦然。如果新消息尚未排队，就会送入消息缓冲区。如果文件系统未加锁，那么新消息将被送入消息缓冲区。新消息不会被送入消息缓冲区。”

37. 你会用什么样的布尔检索来寻找关于新泽西州海滩的网页？如果你想找关于泽西岛（在英吉利海峡）海滩的网页呢？

38. 你会用什么样的布尔检索来寻找关于徒步旅行西弗吉尼亚的网页？如果你想找关于徒步旅行弗吉尼亚的网页，而不是西弗吉尼亚呢？

练习 39～42 是智力游戏题，解题时可以先把语句翻译成逻辑表达式，再用真值表从这些表达式作推理。

39. 斯蒂夫想用两个事实来判断三位工作伙伴的相对薪水。首先他知道如果傅雷德的薪水不

是三人中最高的，那么杰尼斯的最高。其次他知道如果杰尼斯的薪水不是最低的，那么麦吉的最高。从以上斯蒂夫知道的事实，有可能决定傅雷德、麦吉和杰尼斯的相对薪水吗？如果能，谁的最高，谁的最低？解释你的推理。

40. 五个朋友都能进入谈话室。如果知道下面这些信息，能决定谁在谈话吗？凯文或希思或他们两个都在谈话。兰迪或维杰在谈话，但没有同时谈话。如果阿比在谈话，那么兰迪也在谈话。维杰和凯文或者两人都在谈话，或者都不谈话。如果希思在谈话，那么阿比和凯文也在谈。解释你的推理。

41. 侦探调查了罪案的四位证人。从证人的话侦探得出的结论是：如果男管家说的是真话，那么厨师说的也是真话；厨师和园丁说的不可能都是真话；园丁和杂役不可能都在说谎；如果杂役说真话，那么厨师在说谎。侦探能判定这四位证人分别是在说谎还是在说真话吗？解释你的推理。

42. 四个朋友被认定为非法进入某计算机系统的嫌疑人。他们已对调查员作了陈述。艾丽斯说“卡罗斯干的。”约翰说“我没干。”卡罗斯说“戴安娜干的。”戴安娜说“卡罗斯说是我干的，他说谎。”

 a) 如果调查员知道四个嫌疑人中恰有一人说真话，那么谁干的？解释你的推理。

 b) 如果调查员知道恰有一人说谎，谁干的？解释你的推理。

1.2　命题等价

1.2.1　引言

数学证明中使用的重要的一类步骤是用真值相同的一个语句取代另一个。因此，在构造数学证明时广泛使用从已知复合命题产生具有同样真值的其他命题的方法。

我们就从根据可能的真值对复合命题进行分类开始讨论。

定义 1　复合命题称为*永真式*（或*重言式*），如果无论其中出现的命题的真值是什么，它的真值总是真。真值永远为假的复合命题称为*矛盾*。最后，既不是永真式又不是矛盾的命题称为*可能式*。

在数学推理中永真式和矛盾往往是重要的，下面的例子解释了这两类命题。

例 1　我们可以只用一个命题构造永真式和矛盾。$p \vee \neg p$ 和 $p \wedge \neg p$ 的真值表如表 1-8 所示。因为 $p \vee \neg p$ 总是真，它是永真式。因为 $p \wedge \neg p$ 总是假，它是矛盾。■

表 1-8　永真式和矛盾的例子

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

1.2.2　逻辑等价

在所有可能的情况下都有相同真值的两个复合命题称为*逻辑等价*。我们也可以如下定义这一概念。

定义 2 如果 $p \leftrightarrow q$ 是永真式，命题 p 和 q 称为是逻辑等价的。记号 $p \Leftrightarrow q$ 表示 p 和 q 逻辑等价。

判定两个命题是否等价的方法之一是使用真值表。特别是，命题 p 和 q 等价当且仅当给出它们真值的两列完全一致。下面例子解释了这一方法。

例 2 证明 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 等价。这一等价关系是德摩根定律之一。这是用 19 世纪中叶英国数学家德摩根（Augustus De Morgan）[㊀]的名字命名的。

解 表 1-9 给出了这些命题的真值。由于对 p 和 q 所有可能的真值组合，命题 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 的真值都一样，这两个命题逻辑等价。

表 1-9 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 的真值表

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

■

例 3 证明命题 $p \rightarrow q$ 和 $\neg p \vee q$ 逻辑等价。

㊀ 奥古斯塔·德摩根（Augustus De Morgan，1806—1871）生于印度，他父亲是印度陆军上校。德摩根 7 个月大时全家移居英国。他上的是私立学校并在那里培养了对数学的兴趣。德摩根 1827 年毕业于剑桥 Trinity 学院。尽管他想过要学医或学法律，最后还是决定以数学为毕生事业。1828 年他获得了伦敦 University College 的一个职位，但当他的一位教授同事被无故解雇时他辞职了。不过，在 1836 年他的继任人去世后他又回到了自己的位置，直到 1866 年。

德摩根以强调原理胜于技术而著名。他的学生中有许多是著名的数学家，包括拉弗雷斯伯爵夫人奥古斯塔（Ada Augusta），她是巴贝奇（Charles Babbage）计算机器研究的合作者（参见关于 Ada Augusta 的生平小注[㊁]）。

德摩根是位特别多产的作家，他为不止 15 家期刊写了 1000 多篇文章。德摩根还为许多学科撰写课本，包括逻辑、概率、微积分和代数。1838 年，他首次给出了数学归纳法这一重要证明技术的清晰解释，数学归纳法这一术语即由他创造。18 世纪 40 年代他对符号逻辑的发展做出了奠基性的贡献。他发明了帮助他证明命题等价的符号，其中包括以他的名字命名的定律。1842 年，德摩根给出了关于极限的也许是第一个准确定义，并提出了无穷数列收敛的若干检验标准。德摩根还对数学史有兴趣，写了牛顿和哈雷的生平传记。

1837 年，德摩根与弗伦德（Sophia Frend）结婚，后者在 1882 年撰写了德摩根传记。德摩根的研究、写作和教学使他无暇顾及家庭和社交，不过他的善良、幽默及广博的知识仍是闻名于世的。

㊁ 拉弗雷斯伯爵夫人（Ada Augusta，1815—1852） 艾达·奥古斯塔是著名诗人拜伦（Byron）勋爵和安娜贝钠·米尔班克（Annabella Millbanke）的唯一孩子，然而他们在艾达 1 个月大时就分居了。艾达由母亲养大，也是母亲发掘了艾达在智力上的天赋。艾达的老师是数学家威廉·弗伦德（William Frend）和奥古斯塔·德摩根（Augustus De Morgan）。1838 年，艾达同金（King）爵士结婚，此人后来被晋升为拉弗雷斯伯爵。他们一共有 3 个孩子。

艾达结婚后仍坚持她在数学上的研究，帮助查理·巴贝奇（Charles Babbage）从事早期的计算机器的研究工作，这种机器称为解析机。关于此种机器最完整的解释可在艾达的著作中找到。1845 年以后，艾达同巴贝奇一同致力于能预测赛马结果的系统的研究。然而，他们的系统不能运行，使艾达去世时还因此欠下大笔债务。程序设计语言 Ada 就是为纪念这位伯爵夫人而命名的。

解 在表 1-10 中构造了这两个命题的真值表。由于$\neg p \vee q$ 和 $p \rightarrow q$ 的真值一致，它们是逻辑等价的。

表 1-10 $\neg p \vee q$ 和 $p \rightarrow q$ 的真值表

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

■

例 4 证明命题 $p \vee (q \wedge r)$和$(p \vee q) \wedge (p \vee r)$逻辑等价。这是析取对合取的分配律。

解 表 1-11 中构造了这两个命题的真值表。因为$p \vee (q \wedge r)$和$(p \vee q) \wedge (p \vee r)$的真值一样，它们是逻辑等价的。

注意 *涉及 3 个不同命题的复合命题的真值表需要 8 行，每一行代表这 3 个命题的真值的一种组合。一般涉及 n 个命题的复合命题需要 2^n 行。*

表 1-11 $p \vee (q \vee r)$和$(p \vee q) \wedge (p \vee r)$逻辑等价的演示

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

■

表 1-12 给出了若干重要的等价关系[㊀]。在这些等价关系中，T 表示永远为真的任何命题，F 表示永远为假的任何命题。本节末的练习中要求读者证明这些等价关系。

析取的结合律表明表达式 $p \vee q \vee r$ 是有定义的。因为先析取 p 和 q 再与 r 析取或先析取 q 和 r 再与 p 析取，其结果一样。同样，$p \wedge q \wedge r$ 也是有定义的。扩展这一推理过程可以证明：$p_1 \vee p_2 \vee \cdots \vee p_n$ 和 $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ 均有定义，只要 p_1，p_2，$\cdots$，p_n 为命题。进而可以注意到德摩根律可以扩展为

$$\neg(p_1 \vee p_2 \cdots \vee p_n) \Leftrightarrow (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

㊀ 这些恒等式是布尔代数中恒等式的一种特殊情况。请将这些恒等式与 1.5 节表 1-17 中的集合恒等式比较一下，也与 9.1 节表 9-5 中的布尔恒等式比较一下。

和

$$\neg(p_1 \wedge p_2 \cdots \wedge p_n) \Leftrightarrow (\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n)$$

证明这些恒等式的方法将在第 3 章给出。

表 1-12 逻辑等价

等价关系	名 称
$p \wedge \mathrm{T} \Leftrightarrow p$	恒等律
$p \vee \mathrm{F} \Leftrightarrow p$	
$p \vee \mathrm{T} \Leftrightarrow \mathrm{T}$	支配律
$p \wedge \mathrm{F} \Leftrightarrow \mathrm{F}$	
$p \vee p \Leftrightarrow p$	幂等律
$p \wedge p \Leftrightarrow p$	
$\neg(\neg p) \Leftrightarrow p$	双非律
$p \vee q \Leftrightarrow q \vee p$	交换律
$p \wedge q \Leftrightarrow q \wedge p$	
$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$	结合律
$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$	
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$	分配律
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	
$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$	德摩根定律
$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$	

表 1-12 中的逻辑等价关系以及业已建立起来的其他（如表 1-13 中所示的那些）等价关系，可以用于构造其他等价关系，原因是复合命题中的一个命题可以用与它逻辑等价的命题替换而不改变复合命题的真值。这种方法可由例 5 和例 6 得到说明。在这两个例子中，我们还使用了如下事实（见练习 40）：如果 p 和 q 逻辑等价，q 和 r 逻辑等价，那么 p 和 r 也逻辑等价。

表 1-13 若干有用的逻辑等价关系

$p \vee \neg p \Leftrightarrow \mathrm{T}$
$p \wedge \neg p \Leftrightarrow \mathrm{F}$
$(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$

例 5 证明 $\neg(p \vee (\neg p \wedge q))$ 和 $\neg p \wedge \neg q$ 逻辑等价。

解 我们可用真值表证明这两个复合命题等价。不过我们不这样做，而是每次使用表 1-12 中的一个等价关系，依次建立一串等价。从 $\neg(p \vee (\neg p \wedge q))$ 开始，到 $\neg(p \wedge \neg q)$ 结束，最终证明它们等价。我们有下列等价关系。

$$\neg(p \vee (\neg p \wedge q)) \Leftrightarrow \neg p \wedge \neg(\neg p \wedge q) \quad \text{由第二德摩根定律}$$

$$\Leftrightarrow \neg p \wedge (\neg(\neg p) \vee \neg q) \quad \text{由第一德摩根定律}$$

$\Leftrightarrow \neg p \wedge (p \vee \neg q)$	由双非律
$\Leftrightarrow (\neg p \wedge p) \vee (\neg p \wedge \neg q)$	由分配律
$\Leftrightarrow \mathrm{F} \vee (\neg p \wedge \neg q)$	由于$\neg p \wedge p \Leftrightarrow \mathrm{F}$
$\Leftrightarrow (\neg p \wedge \neg q) \vee \mathrm{F}$	由析取的交换律
$\Leftrightarrow \neg p \wedge \neg q$	由 F 的恒等律

于是$\neg(p \vee (\neg p \wedge q))$和$\neg p \wedge \neg q$ 逻辑等价。 ■

例 6 证明$(p \wedge q) \to (p \vee q)$为永真式。

解 为证明这个命题是永真式，我们将用逻辑等价证明它逻辑上等价于 T。（注意：这也可以用真值表来完成。）

$(p \wedge q) \to (p \vee q) \Leftrightarrow \neg(p \wedge q) \vee (p \vee q)$	由例 3
$\Leftrightarrow (\neg p \vee \neg q) \vee (p \vee q)$	由第一德摩根定律
$\Leftrightarrow (\neg p \vee p) \vee (\neg q \vee q)$	由析取的结合律和交换律
$\Leftrightarrow \mathrm{T} \vee \mathrm{T}$	由例 1 和析取的交换律
$\Leftrightarrow \mathrm{T}$	由支配律 ■

真值表可以用于判定复合命题是否为永真式。对于只含少数变量的命题，可以用手工完成这一工作。但当变量数目增长时，就不可行了。例如，对于含 20 个变量的命题，它的真值表就有 $2^{20} = 1\,048\,576$ 行。显然，你需要一台计算机帮助你以这种方式判定含 20 个变量的命题是否为永真式。但是当变量数为 1 000 时，一台计算机能在一个可以接受的时间内判定复合命题是否为永真式吗？要检查 $2^{1\,000}$种（这是一个超过 300 位的十进制数）可能的真值组合中的每一种，一台计算机在几万亿年之内都不可能完成。而且迄今尚没有其他已知的计算过程能使计算机在合理的时间之内判定变量数这么大的命题是否为永真式。在第 2 章我们学习算法复杂性时，将研究这一类问题。

练习

1. 用真值表证明下列等价关系。
   a) $p \wedge \mathrm{T} \Leftrightarrow p$　　b) $p \vee \mathrm{F} \Leftrightarrow p$
   c) $p \wedge \mathrm{F} \Leftrightarrow \mathrm{F}$　　d) $p \vee \mathrm{T} \Leftrightarrow \mathrm{T}$
   e) $p \vee p \Leftrightarrow p$　　f) $p \wedge p \Leftrightarrow p$
2. 证明$\neg(\neg p)$ 和 p 逻辑等价。
3. 用真值表证明交换律。
   a) $p \vee q \Leftrightarrow q \vee p$
   b) $p \wedge q \Leftrightarrow q \wedge p$
4. 用真值表证明结合律
   a) $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
   b) $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
5. 用真值表证明分配律。
   $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
6. 用真值表证明等价关系。

$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$

7. 用真值表证明下列各蕴含关系为永真式。

   a) $(p \wedge q) \rightarrow p$　　b) $p \rightarrow (p \vee q)$

   c) $\neg p \rightarrow (p \rightarrow q)$　　d) $(p \wedge q) \rightarrow (p \rightarrow q)$

   e) $\neg(p \rightarrow q) \rightarrow p$　　f) $\neg(p \rightarrow q) \rightarrow \neg q$

8. 用真值表证明下列蕴含关系为永真式。

   a) $[\neg p \wedge (p \vee q)] \rightarrow q$

   b) $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$

   c) $[p \wedge (p \rightarrow q)] \rightarrow q$

   d) $[(p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)] \rightarrow r$

9. 不用真值表证明练习 7 中的各蕴含关系为永真式。

10. 不用真值表证明练习 8 中的各蕴含关系为永真式。

11. 证明下列称为吸收律的等价关系。

   a) $[p \vee (p \wedge q)] \Leftrightarrow p$

   b) $[p \wedge (p \vee q)] \Leftrightarrow p$

12. 判断 $(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$ 是否为永真式。

13. 判断 $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$ 是否为永真式。

14. 证明 $p \leftrightarrow q$ 和 $(p \wedge q) \vee (\neg p \wedge \neg q)$ 等价。

15. 证明 $(p \rightarrow q) \rightarrow r$ 和 $p \rightarrow (q \rightarrow r)$ 不等价。

16. 证明 $p \rightarrow q$ 和 $\neg q \rightarrow \neg p$ 逻辑等价。

17. 证明 $\neg p \leftrightarrow q$ 和 $p \leftrightarrow \neg q$ 逻辑等价。

18. 证明 $\neg(p \oplus q)$ 和 $p \leftrightarrow q$ 逻辑等价。

19. 证明 $\neg(p \leftrightarrow q)$ 和 $\neg p \leftrightarrow q$ 逻辑等价。

只含逻辑运算符 ∨、∧ 和 ¬ 的复合命题的对偶是将该命题中的每个 ∨ 用 ∧ 代替，每个 ∧ 用 ∨ 代替，每个 T 用 F 代替，每个 F 用 T 代替得到的命题。命题 s 的对偶用 s^* 表示。

20. 求下列命题的对偶。

   a) $p \wedge \neg q \wedge \neg r$　　b) $(p \wedge q \wedge r) \vee s$

   c) $(p \vee \mathrm{F}) \wedge (q \vee \mathrm{T})$

21. 证明 $(s^*)^* = s$

22. 表 1-12 中的逻辑等价关系除双非律外都是成对的。证明每一对命题都是互为对偶的。

**23. 为什么两个只含运算符 ∧、∨ 和 ¬ 的等价复合命题的对偶也等价？

24. 找一个只含命题 p、q 和 r 的复合命题，当 p 和 q 为真而 r 为假时命题为真，否则为假。[提示：用各个命题或其否定构造合取。]

25. 找一个只含命题 p、q 和 r 的复合命题，在 p、q 和 r 中恰有两个为真时命题为真，否则为假。[提示：构造合取的析取。对命题成真的每一种组合包含一个合取。每个合取都应包含三个命题中每个命题或它的否定。]

26. 假定用 n 个命题变量给出一个真值表。证明可依此表构造一个复合命题，使其真值与此表一致。构造办法是：对变量的每一种使复合命题成真的组合，用变量或变量的否定作

成合取；再对所有合取作析取。这样得到的复合命题称为析取范式。

如果每一个复合命题都逻辑等价于一个只含某些运算符的复合命题，则这一组逻辑运算符称为功能完备的。

27. 证明¬、∧和∨构成一个功能完备的逻辑运算符集合。[提示：利用练习 26 中给出的事实，即每个命题都逻辑等价于一个析取范式。]

*28. 证明¬ 和∧构成一个功能完备的逻辑运算符集合。[提示：首先用德摩根定律证明 $p \vee q$ 等价于 $\neg(\neg p \wedge \neg q)$。]

*29. 证明¬ 和∨构成一个功能完备的逻辑运算符集合。

下面几道题用到逻辑运算符 *NAND*（与非）和 *NOR*（或非）。命题 p *NAND* q 在 p 为假或 q 为假或两者均为假时为真；当 p 和 q 均为真时为假。命题 p *NOR* q 只在 p 和 q 均为假时为真，否则为假。命题 p *NAND* q 和 p *NOR* q 分别表示为 $p \mid q$ 和 $p \downarrow q$。(运算符 | 和 ↓ 分别以谢非尔和皮尔斯的名字命名为谢菲尔竖和皮尔斯[㊀]箭头。)

30. 为运算符 *NAND* 构造真值表。

31. 证明 $p \mid q$ 逻辑等价于 $\neg(p \wedge q)$。

32. 为运算符 *NOR* 构造真值表。

33. 证明 $p \downarrow q$ 逻辑等价于 $\neg(p \vee q)$。

34. 本题将证明 {↓} 是功能完备的一个逻辑运算符集合。

   a) 证明 $p \downarrow p$ 逻辑等价于 $\neg p$。

   b) 证明 $(p \downarrow q) \downarrow (p \downarrow q)$ 逻辑等价于 $p \vee q$。

   c) 从练习 29 和本题的 a)、b) 得知 {↓} 是功能完备的一个逻辑运算符集合。

*35. 只用运算符↓构造一个等价于 $p \to q$ 的命题。

㊀ 查理·皮尔斯（Charles Sanders Peirce，1839—1914）　许多人都认为查理·皮尔斯是美国最有创造性和最多才多艺的知识分子。他生于麻省的剑桥，其父本杰明·皮尔斯是哈佛大学的数学和自然哲学教授。皮尔斯就学于哈佛（1855—1859 年）并获艺术硕士学位（1862），后在劳伦斯科学学校获化学高级学位（1863）。他父亲鼓励他从事自然科学，但他却选择了研究逻辑和科学方法论。

1861 年，为更好理解科学方法论，皮尔斯当上了美国海岸观测署的助理。他在观测署的服务使他在南北战争期间免于从军。在该署工作期间，皮尔斯进行了天文和大地测量工作。他应用椭圆函数理论的最新数学成果，对钟摆设计和地图投影做出了奠基性的贡献。他是第一个把光的波长做为度量单位的人。皮尔斯被提升为观测署助理署长，并担任此职直到 1891 年被迫退职，因为他不同意观测署新的行政当局设定的工作方向。

尽管皮尔斯毕生致力于物理科学，他仍然提出了各种科学的一个层次结构，其中数学位于最高层，而且一门科学的方法可以被其下层科学采用。他还是美国实用主义哲学理论的奠基人。

皮尔斯唯一的学术职位是 1879—1884 年在巴尔的摩的约翰斯·霍普金斯大学担任逻辑学讲师。这期间他完成的数学工作包括他对逻辑、集合论、抽象代数和数学原理的贡献。他的工作至今仍产生影响，他在逻辑上的某些工作近来已被应用于人工智能。皮尔斯相信，研究数学可以开发大脑的想象力、抽象思维能力和归纳能力。他从观测署退休以后的五花八门的工作包括为报纸和期刊写稿、编著多部学术辞典、翻译科技论文、客座授课及撰写教科书。不幸的是，以这些工作得到的收入不足以使他和他的第二任妻子免受贫穷之苦。晚年他得到由他的许多赞赏者创立的基金的支持，该基金由他终生的朋友哲学家威廉·詹姆斯（Willian James）管理。尽管皮尔斯就广泛的主题写作并出版了大量著作，他仍然留下了 100 000 多页未出版的手稿。由于这些未发表的作品很难读，学者们只是在近来才开始理解他大量贡献中的一部分。一些人正献身于把他的工作推上因特网，从而使皮尔斯对全世界的贡献更好地得到鉴赏。

36. 证明 $\{|\}$ 是功能完备的一个逻辑运算符集合。

37. 证明 $p|q$ 和 $q|p$ 等价。

38. 证明 $p|(q|r)$ 和 $(p|q)|r$ 不等价。（因此，逻辑运算符 $|$ 不满足结合律。）

*39. 只用命题 p 和 q 能有多少不同的复合命题真值表？

40. 如果复合命题 p、q 和 r 中，p 与 q 逻辑等价，q 与 r 逻辑等价，证明 p 与 r 逻辑等价。

41. 下面的语句取自一个电话系统的规范："如果电话号码数据库是打开的，那么监督程序被置于关闭状态，只要系统不在初态。"这句话有两个蕴含，使规范很难懂。找一个等价的易懂的规范，使其只涉及析取和否定，不涉及蕴含。

1.3 谓词和量词

1.3.1 引言

含变量的语句，如

"$x>3$"，"$x=y+3$" 和 "$x+y=z$"

常见于数学断言和计算机程序。在变量值未知的时候，这些语句既不成真也不为假。本节将讨论从这种语句产生命题的方式。

语句"x 大于 3"有两部分，第一部分即变量 x 是语句的主语，第二部分谓词"大于 3"表明语句的主语会有的一个性质。我们可以用 $P(x)$ 表示语句"x 大于 3"，其中 P 表示谓词"大于 3"，而 x 是变量。也把语句 $P(x)$ 说成是*命题函数* P 在 x 的值。一旦给变量 x 赋一个值，语句 $P(x)$ 就成为命题，因而有真值。考虑下面的例子。

例 1　令 $P(x)$ 表示语句"$x>3$"，$P(4)$ 和 $P(2)$ 的值是什么？

解　在语句"$x>3$"中让 $x=4$ 即得到语句 $P(4)$。因此 $P(4)$ 即语句"$4>3$"，为真；但语句 $p(2)$ 即"$2>3$"，为假。■

还可以让语句中含不止一个变量，例如，考虑语句"$x=y+3$"。我们可以用 $Q(x,y)$ 表示这个语句，其中 x，y 为变量，Q 为谓词。在赋值给 x 和 y 时，语句 $Q(x,y)$ 就有了真值。

例 2　令 $Q(x,y)$ 表示语句"$x=y+3$"，命题 $Q(1,2)$ 和 $Q(3,0)$ 的值是什么？

解　要得到 $Q(1,2)$，在 $Q(x,y)$ 中令 $x=1$，$y=2$。因此，$Q(1,2)$ 为语句"$1=2+3$"，为假。语句 $Q(3,0)$ 为语句"$3=0+3$"，为真。■

同样，可以用 $R(x,y,z)$ 表示语句"$x+y=z$"，在 x、y 和 z 被赋值时，这一语句就有了真值。

例 3　命题 $R(1,2,3)$ 和 $R(0,0,1)$ 的真值是什么？

解　在语句 $R(x,y,z)$ 中令 $x=1$，$y=2$，$z=3$，即得到命题 $R(1,2,3)$。可以看出 $R(1,2,3)$ 就是语句"$1+2=3$"，为真。$R(0,0,1)$ 即语句"$0+0=1$"，为假。■

一般涉及 n 个变量如 x_1，x_2，…，x_n 的语句可以用 $P(x_1,x_2,\cdots,x_n)$ 表示。

形为$P(x_1,x_2,\cdots,x_n)$的语句是命题函数 P 在 n 元组（x_1，x_2，…，x_n）的值，P 也称为谓词。

下面的例子说明，命题函数可以出现在计算机程序中。

例 4 考虑语句

```
if x>0 then x := x+1
```

当程序中遇到这样一条语句时，变量 x 在程序运行到此刻的值即被代入$P(x)$，也就是代入"$x>0$"。如果对 x 的这一值$P(x)$为真，即执行赋值语句 $x := x+1$，于是 x 的值增加1。如果对 x 的这一值$P(x)$为假，则不执行赋值语句，所以 x 的值不改变。 ■

1.3.2 量词

当命题函数中所有变量均被赋值时，得到的命题有一个真值。还有另一重要方式，也可以从命题函数产生命题，这就是量化。本节将讨论两类量化，即全称量化和存在量化。

许多数学语句断定某一性质对变量在某一特定域内的所有值均为真，这一特定域称为变量的论域。这类语句用全称量化表示。命题函数的全称量化是这样一个命题：它断言$P(x)$对 x 在其论域中的所有值为真，论域规定变量 x 可能取的值。

定义 1 $P(x)$的全称量化是命题"$P(x)$对 x 在其论域的所有值为真"。

符号

$$\forall x P(x)$$

表示$P(x)$的全称量化，其中 $\forall$ 称为全称量词。命题 $\forall x P(x)$也可表示为"对所有 x，$P(x)$"或"对每个 x，$P(x)$"。

注意 最好不用"任何"一词，因为它常引起歧义，不知指的是"每一个"还是"某一个"。在有些情况下，"任何"没有二义性，例如，用于否定句时，像"没有任何理由不努力学习。"

例 5 用全称量化表示语句

"本班每一个学生都已学过微积分。"

解 用$P(x)$表示语句

"x 已学过微积分。"

那么语句"本班每一个学生都已学过微积分"可以写成 $\forall x P(x)$，其中的论域由本班学生组成。

这一语句也可以表示为

$$\forall x(S(x)\rightarrow P(x))$$

其中$S(x)$是语句

"x 属于本班。"

$P(x)$与上面相同，而论域则是所有学生的集合。 ■

例 5 说明，往往有不止一种好方式表示同一全称量化。

例 6 令 $P(x)$ 为语句"$x+1>x$"，量化语句 $\forall xP(x)$ 的真值是什么？其中论域是实数集合。

解 由于 $P(x)$ 对所有实数 x 均为真，量化语句

$$\forall xP(x)$$

的值为真。■

例 7 令 $Q(x)$ 表示语句"$x<2$"。若论域是实数集合，量化语句 $\forall xQ(x)$ 的真值是什么？

解 $Q(x)$ 并非对所有实数都为真，例如，$Q(3)$ 就为假，因此

$$\forall xQ(x)$$ ■

为假。

当论域中的所有元素可以一一列出——即例如列成 x_1，x_2，…，x_n——时，量化语句 $\forall xP(x)$ 与合取

$$P(x_1)\wedge P(x_2)\wedge\cdots\wedge P(x_n)$$

是一回事，因为这一合取为真当且仅当 $P(x_1),P(x_2),\cdots,P(x_n)$ 全为真。

例 8 若论域是不超过 4 的正整数，$P(x)$ 是语句"$x^2<10$"，$\forall xP(x)$ 的真值是什么？

解 语句 $\forall xP(x)$ 就是合取

$$P(1)\wedge P(2)\wedge P(3)\wedge P(4)$$

因为论域由 1、2、3 和 4 组成。由于 $P(4)$ 即语句"$4^2<10$"为假，$\forall xP(x)$ 为假。■

许多数学语句断定有一个具有某种性质的元素。这类语句用存在量化表示。用存在量化可以构成这样一个命题：它成真的充分必要条件是论域中至少有一个值使 $P(x)$ 为真。

定义 2 *$P(x)$ 的存在量化是命题"论域中存在一个元素 x 使 $P(x)$ 为真"。*

我们用符号

$$\exists xP(x)$$

表示 $P(x)$ 的存在量化，其中 $\exists$ 称为*存在量词*。存在量化 $\exists xP(x)$ 也可表示为

"有一个 x 使得 $P(x)$，"

"至少有一个 x 使得 $P(x)$，"

或

"对某个 x，$P(x)$。"

例 9 令 $P(x)$ 表示语句"$x>3$"，论域为实数集合，量化语句 $\exists xP(x)$ 的真值是什么？

解　因为 “$x>3$” 在如 $x=4$ 时为真，$P(x)$的存在量化 $\exists xP(x)$为真。■

例 10　令$Q(x)$表示语句 “$x=x+1$”，论域是实数集，量化语句 $\exists xQ(x)$的真值是什么？

解　因为对每个实数 x，$Q(x)$都为假，$Q(x)$的存在量化 $\exists xQ(x)$为假。■

当论域中的所有元素可以一一列出——例如列成 x_1，x_2，…，x_n——时，存在量化 $\exists xP(x)$与析取

$$P(x_1)\vee P(x_2)\vee\cdots\vee P(x_n)$$

是一回事，因为这一析取当且仅当$P(x_1),P(x_2),\cdots,P(x_n)$中至少一个为真时为真。

例 11　若 $P(x)$是语句 “$x^2>10$”，论域为不超过 4 的正整数，$\exists xP(x)$的真值是什么？

解　由于论域为 {1，2，3，4}，命题 $\exists xP(x)$就是析取

$$P(1)\vee P(2)\vee P(3)\vee P(4)$$

由于$P(4)$即 “$4^2>10$” 为真，$\exists xP(x)$为真。■

表 1-14 总结了全称量词和存在量词的含义。

表 1-14　量词

语　句	何时为真	何时为假
$\forall xP(x)$	对每一个 x，$P(x)$都为真	有一个 x，使 $P(x)$为假
$\exists xP(x)$	有一个 x，使 $P(x)$为真	对每一个 x，$P(x)$都为假

在决定量化语句的真值时，借助循环与搜索来思考是有益的。假定在变量 x 的论域中有 n 个对象，要决定 $\forall xP(x)$是否为真，可以对 x 的 n 个值循环查看$P(x)$是否总是真。如果遇到 x 的一个值使$P(x)$为假，那就证明 $\forall xP(x)$为假，否则 $\forall xP(x)$为真。要决定 $\exists xP(x)$是否为真，我们循环查看 x 的 n 个值，搜索使$P(x)$成真的 x 之值。如果找到一个，那么 $\exists xP(x)$为真；如果总也找不到这样的 x，则判定 $\exists xP(x)$为假。(注意，当论域有无穷多个值时，这一搜索过程不适用。不过以这种方式思考量化语句的真值仍是有益的。)

有时含量词的表达式会相当复杂，把复杂的表达式翻译成文字语句有助于对其含义的理解。翻译的第一步是写出每个量词的含义，然后是用简化的句子表达这一含义。

例 12　把语句

$$\forall x(C(x)\vee\exists y(C(y)\wedge F(x,y)))$$

译成文字语句，其中$C(x)$是 “x 有台计算机”，$F(x,y)$是 “x 和 y 是朋友”，而 x 和 y 的共同论域是学校全体学生的集合。

解　该语句说的是，对学校的每一个学生 x，或者 x 有台计算机，或者另有学生 y，他有台计算机，并且 x 和 y 是朋友。换言之，学校的每个学生或有计算机，或有个有计算机的朋友。■

例 13 把语句

$$\exists x \forall y \forall z((F(x,y) \land F(x,z) \land (y \neq z)) \to \neg F(y,z))$$

译成文字语句，其中$F(a,b)$的含义是 a 和 b 是朋友，而 x、y 和 z 的论域是学校所有学生的集合。

解 这个语句说的是，有一个学生 x，对所有学生 y 及不同于 y 的所有学生 z，只要 x 和 y 是朋友，x 和 z 也是朋友，那么 y 和 z 就不是朋友。换句话说，有个学生，他的朋友之间都不是朋友。■

包含量词的复杂表达式也出现在数学语句中，下面就是一个例子。

例 14 假定变量 x 和 y 的论域是所有实数的集合，语句

$$\forall x \forall y(x + y = y + x)$$

说的是对所有实数 x 和 y，$x + y = y + x$。这是实数加法的交换律。同样，语句

$$\forall x \exists y(x + y = 0)$$

说的是对所有实数 x，有一个实数 y，使得 $x + y = 0$。也就是每个实数都有一个加法的逆。同样，语句

$$\forall x \forall y \forall z(x + (y + z) = (x + y) + z)$$

是实数加法的结合律 ■

1.3.3 翻译语句为逻辑表达式

在 1.1 节中我们阐明了把句子译成由命题和逻辑联结词构成逻辑表达式的过程。现在我们已经讨论了量词，因而能把更广泛的句子译成逻辑表达式。这样做既可以去掉二义性，又使得用这些句子推理成为可能。(3.1 节讲述了用逻辑表达式推理的推理规则。)

下面的几个例子说明如何用逻辑运算符和量词表达句子。这是一些常出现在数学陈述、逻辑程序设计和人工智能中的语句。

例 15 用量词表达语句“这个班上某个学生去过墨西哥”和“这个班上每个学生或去过加拿大，或去过墨西哥。”

解 令变量 x 的论域为这个班所有学生的集合，令$M(x)$为语句“x 去过墨西哥”，$C(x)$为语句“x 去过加拿大”。语句“这个班上某个学生去过墨西哥”可表示为$\exists x M(x)$。语句“这个班上每个学生或去过加拿大，或去过墨西哥”可以写成$\forall x(C(x) \lor M(x))$(这里假定是同或意义上的或)。■

例 16 把语句“每人恰有一个最好的朋友”表示为逻辑表达式。

解 令$B(x,y)$为语句“y 是 x 的最好朋友”。注意本例中的句子说的是，对每个 x 都有另一人 y 是 x 的最好朋友，而且如果 z 是不同于 y 的另一人，则 z 不是 x 的最好朋友。于是可以把这个语句翻译为

$$\forall x \exists y \forall z(B(x,y) \land ((z \neq y) \to \neg B(x,z)))$$ ■

例 17 把语句“如果某人是女性而且有子女，那么此人一定是某人的母亲”表示为逻辑表达式。

解　令 $F(x)$ 为语句"x 是女性"，$P(x)$ 为语句"x 有子女"，再令 $M(x,y)$ 为语句"x 是 y 的母亲"。由于本例中的语句适合于所有的人，我们可以用符号写成

$$\forall x((F(x) \wedge P(x)) \rightarrow \exists y M(x,y))$$ ■

例 18　用量词表示语句"有位妇女已搭乘过世界上每一条航线上的航班"。

解　令 $P(w,f)$ 为"w 搭乘过 f"，而 $Q(f,a)$ 为"f 是 a 上的航班"。于是可将上述语句表示为

$$\exists w \forall a \exists f(P(w,f) \wedge Q(f,a))$$

其中，w、f 和 a 的论域分别为世界上所有妇女、所有空中航班和所有航线。

这个语句也可以表示为

$$\exists w \forall a \exists f R(w,f,a)$$

其中 R（w，f，a）为"w 已搭乘过 a 上的 f"。虽然这样表示更紧凑，但它使变量之间的关系有点含糊不清，因此，第一个解要好些。 ■

正如前面提到的，量词常用于数学概念的定义，你可能熟悉的极限概念即为一例。极限是微积分中的重要概念。

例 19　（需要微积分知识）用量词表示极限的定义。

解　回顾一下语句

$$\lim_{x \to a} f(x) = L$$

的定义是：对每个实数 $\epsilon>0$，存在一个实数 $\delta>0$，使得对每一个 x，只要 $0<|x-a|<\delta$，就有 $|f(x)-L|<\epsilon$。极限的这一定义用量词表示为

$$\forall \epsilon \exists \delta \forall x(0<|x-a|<\delta \rightarrow |f(x)-L|<\epsilon)$$

其中 ϵ 和 δ 的论域是正实数集合，x 的论域是实数集合。

这一定义还可表示为

$$\forall \epsilon>0 \exists \delta>0 \forall x(0<|x-a|<\delta \rightarrow |f(x)-L|<\epsilon)$$

其中 ϵ 和 δ 的论域为实数集合，而不是正实数集合。 ■

1.3.4　选自 Lewis Carroll 的例子（选读）

Lewis Carroll（实际上是 C.L.Dodgson[⊖]的笔名）是《*Alice in Wonderland*》的作者，也是有关符号逻辑几篇论文的作者。他的书中有用量词推理的例子。下面的两个例子选自他的



⊖ 道奇森（Charles Lutwidge Dodgson，1832—1898）　我们知道 Charles Dodgson 即是 Lewis Carroll，后者是他发表逻辑著作的笔名。道奇森是牧师的儿子，在 11 个子女中排行第三。这 11 个全是结巴。在成年人的包围下他常感到不安。道奇森十分严肃和忠于宗教信仰。他与 Dean Liddell 的三个女儿的友谊使他写成《*Alice in Wonderland*》，这本书为他赢得了金钱和名声。

道奇森 1854 年毕业于牛津，并于 1857 年获得艺术硕士学位。1855 年他被任命为牛津 Christ Church 学院的数学讲师。1861 年英国教会委任他以圣职，但他从来没有实际履行过宗教职责。他的著作包括有关几何、行列式以及竞赛和选举中的数学问题的论文和书籍。（他还以笔名 Lewis Carroll 写过许多关于消遣性逻辑的作品。）

书《*Symbolic logic*》，选自这本书的其他例子在本节末的练习中给出。这些例子说明了怎样用量词表示各种类型的词句。

例 20 考虑下面这些语句，其中头两句称为前提，第三句称为结论。作为一个整体它们被称为一个论证。

"所有狮子都是凶猛的。"
"有些狮子不喝咖啡。"
"有些凶猛的动物不喝咖啡。"

(3.1 节将讨论判定结论部分是不是前提部分有效后果的问题，就这个例子来说，结论是有效的。) 令$P(x)$,$Q(x)$和 $P(x)$分别为语句"x 是狮子"，"x 是凶猛的"和"x 喝咖啡"。假定所有动物的集合为论域，用量词及$P(x)$、$Q(x)$和 $R(x)$表示上面这些语句。

解 可以将这些句子表示为：

$$\forall x(P(x)\rightarrow Q(x))$$
$$\exists x(P(x)\wedge\neg R(x))$$
$$\exists x(Q(x)\wedge\neg R(x))$$

注意，第二句不能表示为$\exists x(P(x)\rightarrow\neg R(x))$。原因是$P(x)\rightarrow\neg R(x)$在 x 不是狮子时总是成真，所以只要有一只不是狮子的动物，$\exists x(P(x)\rightarrow\neg R(x))$就成真，即使所有狮子都喝咖啡它也成真。同样，第三句也不能写成

$$\exists x(Q(x)\rightarrow\neg R(x))$$ ■

例 21 考虑下面的语句，其中前 3 个语句为前提，第 4 个语句为有效结论。

"所有鸣鸟都五彩斑斓。"
"没有大鸟以蜜为生。"
"不以蜜为生的鸟都色彩单调。"
"鸣鸟都是小鸟。"

令$P(x)$,$Q(x)$,$R(x)$和 $S(x)$分别为语句"x 是只鸣鸟"，"x 是大的"，"x 以蜜为生"和"x 五彩斑斓"。假定以所有鸟的集合为论域，用量词及$P(x)$、$Q(x)$、$R(x)$和 $S(x)$表示上述语句。

解 可以把论证中的语句表示为

$$\forall x(P(x)\rightarrow S(x))$$
$$\neg\exists x(Q(x)\wedge R(x))$$
$$\forall x(\neg R(x)\rightarrow\neg S(x)$$
$$\forall x(P(x)\rightarrow\neg Q(x))$$

(注意，我们假定了"小"就是"不大"，"色彩单调"就是"不五彩斑斓"。为证明第 4 句是前 3 句的有效结论，需要用到将在 3.1 节讨论的推理规则。) ■

1.3.5 绑定变量

当量词作用于变量 x 或给这一变量赋值时，我们说此变量的这一次出现为绑定的。没

有被量词绑定或设置为与某一特定值相等的变量出现称为自由的。出现在命题函数中的所有变量必须是绑定的，才能把此命题函数转变为命题。可以用全称量词、存在量词和赋值来完成转变。

许多数学语句需要对多变量命题函数作多重量化。除非所有量词均为全称量词或均为存在量词，否则量词的顺序是重要的。例 22、23 和 24 用于说明这一点。这几个例子中每个变量的论域都假定为实数集合。

例 22　令 $P(x,y)$ 为语句 "$x+y=y+x$"，量化语句 $\forall x\,\forall y\,P(x,y)$ 的真值是什么？

解　量化语句

$$\forall x\,\forall y\,P(x,y)$$

表示的命题是

"对所有实数 x 和所有实数 y，$x+y=y+x$ 成立。"

因为 $P(x,y)$ 对所有实数 x 和 y 成真，$\forall x\,\forall y\,P(x,y)$ 成真。 ■

例 23　令 $Q(x,y)$ 表示 "$x+y=0$"，量化语句 $\exists y\,\forall x Q(x,y)$ 和 $\forall x\,\exists y Q(x,y)$ 的真值是什么？

解　量化语句

$$\exists y\,\forall x Q(x,y)$$

表示的命题是

"有个实数 y 能使 $Q(x,y)$ 对每一个实数 x 成立。"

不管 y 取什么值，只有一个 x 的值能使 $x+y=0$ 成立。因为没有实数 y 能使 $x+y=0$ 对所有实数 x 成立，语句 $\exists y\,\forall x Q(x,y)$ 为假。

量化语句

$$\forall x\,\exists y Q(x,y)$$

表示的命题是

"对每个实数 x 都有一个实数 y 使 $Q(x,y)$ 成立。"

给定一个实数 x，总有一个实数 y 能使 $x+y=0$，这个实数就是 $y=-x$。因此，语句 $\forall x\,\exists y Q(x,y)$ 为真。 ■

例 23 说明量词出现的顺序不同含义也不同。语句 $\exists y\,\forall x P(x,y)$ 和 $\forall x\,\exists y\,P(x,y)$ 不是逻辑等价的，语句 $\exists y\,\forall x P(x,y)$ 成真当且仅当存在一个 y，使得 $P(x,y)$ 对每个 x 都成立。因此，要使这一语句为真，必须有一个特定的 y 值，对这个值无论是什么样的 x，$P(x,y)$ 都成立；另一方面，$\forall x\,\exists y\,P(x,y)$ 为真当且仅当对 x 的每一个值都有一个 y 的值使 $P(x,y)$ 成立。所以，要使这个语句为真，不管你选什么 x，总有 y 的一个值（也许依赖于你选的 x）使 $P(x,y)$ 成立。换言之，在第二种情况下，y 随着 x 而定，但在第一种情况下，y 是与 x 无关的常数。

从这些观察可以看出，如果 $\exists y\,\forall x P(x,y)$ 为真，则 $\forall x\,\exists y\,P(x,y)$ 必定也为真；可是如果 $\forall x\,\exists y\,P(x,y)$ 为真，$\exists y\,\forall x P(x,y)$ 不一定为真。（参见本章补充练习 8 和 10。）

对多个变量使用量词时，借助嵌套循环来思考是有益的。（当然，如果某个变量的论域有无穷多个元素，是无法真正对所有值做循环的。不过这种思考方式对理解嵌套量词总是有益的。）例如，要决定 $\forall x\,\forall y\,P(x,y)$是否为真，我们先对 x 的所有值做循环，而对 x 的每个值再对 y 的所有值循环。如果我们发现对 x 和 y 的所有值$P(x,y)$都为真，那么我们就判定了 $\forall x\,\forall y\,P(x,y)$为真。只要我们碰上一个 x 值，对这个值又有一个 y 值使$P(x,y)$为假，那么就证明了 $\forall x\,\forall y\,P(x,y)$为假。

同样，要判定 $\forall x\,\exists y\,P(x,y)$是否为真，我们对 x 的所有值循环，对 x 的每个值，对 y 的值循环直到找到一个 y 使$P(x,y)$为真。如果对 x 的所有值，我们都能碰上这样的一个 y 值，那么 $\forall x\,\exists y\,P(x,y)$为真。如果对某个 x 我们碰不上这样的 y，那么 $\forall x\,\exists y\,P(x,y)$就为假。

要决定 $\exists x\,\forall y\,P(x,y)$是否为真，需要对 x 的值循环直到找到某个 x，就这个 x 对 y 的所有值循环时$P(x,y)$总是成真。如果能找到这样的 x，$\exists x\,\forall y\,P(x,y)$就为真。如果总也碰不上这样的 x，那么我们知道 $\exists x\,\forall y\,P(x,y)$为假。

最后要看 $\exists x\,\exists y\,P(x,y)$是否为真。我们对 x 的值循环，循环时对 x 的每个值都对 y 的值循环，直到找到 x 的一个值和 y 的一个值使$P(x,y)$为真。只有当我们永远碰不上这样的 x 和 y 能使$P(x,y)$成真时，语句 $\exists x\,\exists y\,P(x,y)$才为假。

表 1-15 是两个变量所有不同的可能量化方式的总结。

例 24 说明，两个以上变量的量化也是常见的。

表 1-15　两个变量的量化

语　句	何时为真	何时为假
$\forall x\,\forall y\,P(x,y)$	对每一对 x、y，$P(x,y)$	有一对 x、y 使 $P(x,y)$
$\forall y\,\forall x P(x,y)$	均为真	为假
$\forall x\,\exists y\,P(x,y)$	对每个 x，都有 y 使 $P(x,y)$为真	有 x，使 $P(x,y)$对每个 y 总是假
$\exists x\,\forall y\,P(x,y)$	有一个 x，使 $P(x,y)$对所有 y 均为真	对每个 x 都有 y 使 $P(x,y)$为假
$\exists x\,\exists y\,P(x,y)$	有一对 x、y 使	对每一对 x，y，
$\exists y\,\exists x P(x,y)$	$P(x,y)$为真	$P(x,y)$均为假

例 24　令$Q(x,y,z)$为语句“$x+y=z$”，语句 $\forall x\,\forall y\,\exists z Q(x,y,z)$和 $\exists z\,\forall x\,\forall y\,Q(x,y,z)$的真值是什么？

解　假定给 x 和 y 赋了值，那么就有一个实数 z，使得 $x+y=z$。于是量化语句

$$\forall x\,\forall y\,\exists z Q(x,y,z)$$

为真，因为它其实就是语句

“对所有实数 x 和所有实数 y，有实数 z，使得 $x+y=z$”。

这里量词出现的顺序是重要的，因为量化语句

$$\exists z\,\forall x\,\forall y\,Q(x,y,z)$$

也就是语句

"有实数 z 使得对所有实数 x 和所有实数 y，$x+y=z$"

为假，因为没有 z 的值能使 $x+y=z$ 对 x 和 y 的所有值都成立。 ■

1.3.6 否定

我们常会考虑量化表达式的否定，例如，考虑语句"班上每个学生都学过一门微积分课"

的否定。这个语句是全称量化语句，即

$$\forall x P(x)$$

其中 $P(x)$ 为语句"x 学过一门微积分课"。这一语句的否定是"并非班上每个学生都学过一门微积分课"。这等价于"班上有个学生没有学过微积分课"。这也就是原命题函数之否定的存在量化，即

$$\exists x \neg P(x)$$

这个例子说明了下面的等价关系：

$$\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$$

假定我们想否定一个存在量化语句。例如，考虑命题"班上有个学生学过一门微积分课。"就是存在量化语句

$$\exists x Q(x)$$

其中 $Q(x)$ 为语句"x 学过一门微积分课"。这句话的否定是命题"并非班上有个学生学过微积分课"。这等价于"班上每个学生都没学过微积分课"，这也就是原命题函数之否定的全称量化，或用量词语言表示为

$$\forall x \neg Q(x)$$

这个例子说明了等价关系

$$\neg \exists x Q(x) \Leftrightarrow \forall x \neg Q(x)$$

表 1-16 给出了量词之否定的总结。

表 1-16 量词的否定

否　定	等价语句	何时为真	何时为假
$\neg \exists x P(x)$	$\forall x \neg P(x)$	对每个 x，$P(x)$ 为假	有 x，使 $P(x)$ 为真
$\neg \forall x P(x)$	$\exists x \neg P(x)$	有 x 使 $P(x)$ 为假	对每个 x，$P(x)$ 为真

练习

1. 令 $P(x)$ 表示语句"$x \leqslant 4$。"下列各项的真值是什么？
   a) $P(0)$　　b) $P(4)$　　c) $P(6)$

2. 令 $P(x)$ 表示语句“单词 x 含字母 a。”下列各项的真值是什么？

a) P(orange)　　b) P(lemon)

c) P(true)　　d) P(false)

3. 令 $Q(x,y)$ 表示语句“x 是 y 的首府。”下列各项的真值是什么？

a) Q(丹佛,科罗拉多)

b) Q(底特律,密歇根)

c) Q(马萨诸塞,波士顿)

d) Q(纽约,纽约)

4. 给出执行语句 if $P(x)$ then $x:=1$ 以后 x 的值，其中 $P(x)$ 为语句“$x>1$”，且执行到上述语句时 x 的值是：

a) $x=0$　　b) $x=1$　　c) $x=2$

5. 令 $P(x)$ 为语句“x 每个工作日都花五个多小时上课”，其中 x 的论域是学生集合。用句子表达下列各量化语句。

a) $\exists x P(x)$　　b) $\forall x P(x)$

c) $\exists x \neg P(x)$　　d) $\forall x \neg P(x)$

6. 令 $P(x,y)$ 表示语句“x 选修 y”，其中 x 的论域是班上全体学生的集合，y 的论域是你校所有计算机科学课程的集合。用句子表达下列各量化语句。

a) $\exists x \exists y P(x,y)$　　b) $\exists x \forall y P(x,y)$

c) $\forall x \exists y P(x,y)$　　d) $\exists y \forall x P(x,y)$

e) $\forall y \exists x P(x,y)$　　f) $\forall x \forall y P(x,y)$

7. 令 $W(x,y)$ 表示“x 访问过 y”，其中 x 的论域是你校全体学生集合，y 的论域是所有网站的集合。用简单的句子表达下列语句。

a) W(Sarah Smith, www.att.com)

b) $\exists x W(x,$ www.imdb.org$)$

c) $\exists y W($Jose Orez$,y)$

d) $\exists y(W($Ashok Puri$,y) \wedge W($Cindy Yoon$,y))$

e) $\exists y \forall z(y \neq ($David Belcher$) \wedge (W($David Belcher$,z) \rightarrow W(y,z)))$

f) $\exists x \exists y \forall z((x \neq y) \wedge (W(x,z) \leftrightarrow W(y,z)))$

8. 令 $C(x,y)$ 表示“x 注册了 y”，其中 x 的论域是你校全体学生的集合，y 的论域是你校开设所有课程的集合。用简单的句子表达下列语句。

a) C(Randy Goldberg, CS 252)

b) $\exists x C(x,$ Math 695$)$

c) $\exists y C($Carol Sitea$,y)$

d) $\exists x(C(x,$ Math 222$) \wedge C(x,$ CS 252$))$

e) $\exists x \exists y \forall z((x \neq y) \wedge (C(x,z) \rightarrow C(y,z)))$

f) $\exists x \exists y \forall z((x \neq y) \wedge (C(x,z) \leftrightarrow C(y,z)))$

9. 令 $P(x)$ 为语句“x 说俄语”，$Q(x)$ 为语句，“x 了解计算机语言 C++”。用 $P(x)$、$Q(x)$、量词和逻辑联结符表示下列各句子，量词的论域为你校全体学生的集合。

a) 你校有个学生既会说俄语又了解 C++。

b）你校有个学生会说俄语，但不了解 C++。
c）你校所有学生或会说俄语，或了解 C++。
d）你校没有学生会说俄语或了解 C++。

10. 令 $Q(x,y)$ 为语句"x 为 y 的参赛者"。用 $Q(x,y)$、量词和逻辑联结符表达下列各句，其中 x 的论域是你校所有学生的集合，y 的论域是所有电视智力竞赛节目。
a）你校有位学生参加了一个电视智力竞赛节目。
b）你校没有学生参加过电视智力竞赛。
c）你校有位学生参加了 Jeopardy 和 Wheel of Fortune 两个电视竞赛。
d）每个电视智力竞赛节目都有你校的一名参赛学生。
e）你校至少有两个学生参加了 Jeopardy 比赛。

11. 令 $L(x,y)$ 为语句"x 爱 y"，其中 x 和 y 的论域都是全世界所有人的集合。用量词表达下列各语句。
a）每个人都爱 Jerry。
b）每个人都爱某个人。
c）有个人人人都爱的人。
d）没有人爱所有的人。
e）有个 Lydia 不爱的人。
f）有个人人都不爱的人。
g）恰有一个人人都爱的人。
h）Lynn 爱的人恰有两个。
i）每个人都爱自己。
j）有人除自己以外谁都不爱。

12. 令 $F(x,y)$ 为语句"x 能骗 y"，其中 x 和 y 的论域为全世界所有人的集合。用量词表达下列语句。
a）人人都能骗 Fred。
b）Evelyn 能骗所有人。
c）每个人都能骗某个人。
d）没有人能骗所有人。
e）人人都会被人骗。
f）没有人能骗 Fred 和 Jerry 两个人。
g）Nancy 恰能骗两个人。
h）恰有一个人人都能骗的人。
i）没有人能骗自己。
j）有人除自己以外只能骗一个人。

13. 令 $S(x)$ 为谓词"x 是学生"，$F(x)$ 为谓词"x 是教员"，而 $A(x,y)$ 是谓词"x 问过 y 问题"，其中论域是你校所有人员的集合。用量词表达下列语句。
a）Lois 问过 Michaels 教授问题。
b）每个学生都问过 Gross 教授问题。
c）每位教员都问过 Miller 教授问题或被 Miller 教授问过问题。

d）某个学生从未问过任何教员的问题。

e）有位教员从未被学生问过问题。

f）有个学生问过所有教员问题。

g）有位教员问过所有其他教员问题。

h）有学生从未被教员问过问题。

14. 令 $I(x)$ 为语句“x 能上因特网”，$C(x,y)$ 为语句“x 和 y 在因特网上交谈过”，其中 x 和 y 的论域是你们班上所有学生的集合。用量词表达下列语句。

a）Jerry 没有上过因特网。

b）Rachel 没在因特网上与 Chelsea 交谈过。

c）Jan 和 Sharon 从未在因特网上交谈过。

d）班上没有人与 Bob 交谈过。

e）除 Joseph 以外，Sanjay 与每个人都交谈过。

f）班上某人没有上过因特网。

g）班上并非人人都上过因特网。

h）班上恰有一人上过因特网。

i）班上除一个学生外都上过因特网。

j）班上上因特网的人在因特网上与班上至少另一名学生交谈过。

k）班上有人上过因特网，但从未与班上其他人交谈过。

l）班上有两个学生没做过网上交谈。

m）班上有个学生与班上每个人都做过网上交谈。

n）班上至少有两个学生没有与同一个人做过网上交谈。

o）班上有两个学生，他们当中有一个与班上其余每个人都交谈过。

15. 令 $M(x,y)$ 为“x 已发给 y 电子邮件”，$T(x,y)$ 为“x 给 y 打过电话”，其中论域为你们班上所有学生。用量词表达下列语句。（假定所有被发出的电子邮件都能收到，尽管事实并非如此。）

a）Chou 从未给 Koko 发过电子邮件。

b）Arlene 从未给 Sarah 发过电子邮件，或未打过电话。

c）Jose 从未收到过 Deborah 的电子邮件。

d）班上每个学生都给 Ken 发过电子邮件。

e）班上没有人给 Nina 打过电话。

f）班上每个人或给 Avi 打过电话或给他发过电子邮件。

g）班上有个学生给班上其他人都发过电子邮件。

h）班上有人给班上其他人或打过电话，或发过电子邮件。

i）班上有两个学生互发过电子邮件。

j）班上有个学生给自己发过电子邮件。

k）班上有个学生既没从班上其他人那里收到过电子邮件，也没有人给他打过电话。

l）班上每个学生都或从同班同学那里收到过电子邮件，或接到过同班同学的电话。

m）班上至少有两个学生，一个给另一个发过电子邮件，第二个则给第一个打过电话。

n）班上有两个同学，他们当中有一个给班上其余同学或发过电子邮件，或打过电话。

16. 用量词表达下列语句。
    a) 班上有个学生会说印地语。
    b) 班上每个学生都会开车。
    c) 班上某学生去过阿拉斯加，但还没去过夏威夷。
    d) 班上所有学生都至少学过一种程序语言。
    e) 班上有个学生已选修学校某个系开设的所有课程。
    f) 班上有个学生恰好与一个同班同学在同一座城市长大。
    g) 班上每个学生都至少与一位同学至少在一个在线聊天室交谈过。
17. 用量词表达下列语句。
    a) 每位学计算机科学的学生都需要上一门离散数学课。
    b) 班上有个学生有一台个人电脑。
    c) 班上每个学生至少选修了一门计算机科学课。
    d) 班上有个学生至少选修过一门计算机科学课。
    e) 班上每个学生都去过校园里每座建筑。
    f) 班上有个学生去过校园里至少一座楼的每个房间。
    g) 班上每个学生都去过校园里每座楼的至少一个房间。
18. 离散数学班上有 1 个主修数学的新生，12 个主修数学的二年级学生，15 个主修计算机科学的二年级学生，2 个主修数学的三年级学生和 1 个主修计算机科学的四年级学生。用量词表达下列语句，再给出其真值。
    a) 班上有个三年级学生。
    b) 班上每个学生都主修计算机科学。
    c) 班上有个学生既不主修数学，也不是三年级学生。
    d) 班上每个学生要么是二年级学生，要么主修计算机科学。
    e) 有一门主修课，有一名学生在每一学年都主修它。
19. 令 $P(x)$ 为语句 "$x = x^2$"。如果论域是整数集合，下列各项的真值是什么？
    a) $P(0)$　　b) $P(1)$　　c) $P(2)$
    d) $P(-1)$　　e) $\exists x P(x)$　　f) $\forall x P(x)$
20. 令 $Q(x,y)$ 为语句 "$x + y = x - y$"。如果两个变量的论域都是整数集合，下列各项的真值是什么？
    a) $Q(1,1)$　　b) $Q(2,0)$
    c) $\forall y\, Q(1,y)$　　d) $\exists x Q(x,2)$
    e) $\exists x \exists y\, Q(x,y)$　　f) $\forall x \exists y\, Q(x,y)$
    g) $\exists y \forall x Q(x,y)$　　h) $\forall y \exists x Q(x,y)$
    i) $\forall x \forall y\, Q(x,y)$
21. 假定所有变量的论域都是整数集合，确定下列语句的真值。
    a) $\forall n(n^2 \geqslant 0)$　　b) $\exists n(n^2 = 2)$
    c) $\forall n(n^2 \geqslant n)$　　d) $\forall n \exists m(n^2 < n)$
    e) $\exists n \forall m(n < m^2)$　　f) $\forall n \exists m(n + m = 0)$
    g) $\exists n \forall m(nm = m)$　　h) $\exists n \exists m(n^2 + m^2 = 5)$

i) $\exists n \exists m (n^2 + m^2 = 6)$
j) $\exists n \exists m (n + m = 4 \wedge n - m = 1)$
k) $\exists n \exists m (n + m = 4 \wedge n - m = 2)$
l) $\forall n \forall m \exists p (p = (m + n)/2)$

22. 假定每个变量的论域都是实数集合，确定下列语句的真值。

a) $\exists x (x^2 = 2)$
b) $\exists x (x^2 = -1)$
c) $\forall x \exists y (x^2 = y)$
d) $\forall x \exists y (x = y^2)$
e) $\exists x \forall y (xy = 0)$
f) $\exists x \exists y (x + y \neq y + x)$
g) $\forall x \neq 0 \exists y (xy = 1)$
h) $\exists x \forall y \neq 0 (xy = 1)$
i) $\forall x \exists y (x + y = 1)$
j) $\exists x \exists y (x + 2y = 2 \wedge 2x + 4y = 5)$
k) $\forall x \exists y (x + y = 2 \wedge 2x - y = 1)$
l) $\forall x \forall y \exists z (z = (x + y)/2)$

23. 假定命题函数 $P(x, y)$ 的论域由 x 和 y 的序偶组成，其中 x 是 1、2 或 3，y 是 1、2 或 3。用析取和合取写出下列命题。

a) $\exists x P(x, 3)$
b) $\forall y P(1, y)$
c) $\forall x \forall y P(x, y)$
d) $\exists x \exists y P(x, y)$
e) $\exists x \forall y P(x, y)$
f) $\forall y \exists x P(x, y)$

24. 重写下列语句，使否定只出现在谓词中（即否定符不在量词外边，也不在含逻辑联结符的表达式外边）。

a) $\neg \exists y \exists x P(x, y)$
b) $\neg \forall x \exists y P(x, y)$
c) $\neg \exists y (Q(y) \wedge \forall x \neg R(x, y))$
d) $\neg \exists y (\exists x R(x, y) \vee \forall x S(x, y))$
e) $\neg \exists y (\forall x \exists z T(x, y, z) \vee \exists x \forall z U(x, y, z))$

25. 重写下列语句，使否定只出现在谓词中（即否定符不在量词外边，也不在含逻辑联结符的表达式外边）。

a) $\neg \forall x \forall y P(x, y)$
b) $\neg \forall x \exists y P(x, y)$
c) $\neg \forall y \forall x (P(x, y) \vee Q(x, y))$
d) $\neg (\exists x \exists y \neg P(x, y) \wedge \forall x \forall y Q(x, y))$
e) $\neg \forall x (\exists y \forall z P(x, y, z) \wedge \exists z \forall y P(x, y, z))$

26. 用量词表达下列语句，然后取该语句的否定并使否定符不在量词的左边。再用简单语句表达这否定（不要简单地表达为“不是…”）。

a) 所有狗都长跳蚤。
b) 没有人因玩彩票损失超过 \$1 000。
c) 班上有个学生只与另外一个学生在网上交谈过。
d) 班上没人恰给另外两个同班同学发过电子邮件。
e) 某个学生已完成本书每道练习。
f) 没有学生做过本书每节至少一道练习。

27. 用量词表达下列语句，然后取该语句的否定并使否定符不在量词左边。再用简单语句表达否定语句（不要简单地表达为“不是…”）。

a) 没有会说话的狗。

b) 班上没人既能说法语又能说俄语。

c) 班上每个学生都恰好选修过本校两门数学课。

d) 有人去过世界上除利比亚以外的每个国家。

e) 没有人攀登过喜马拉雅山的每座山峰。

f) 每位电影演员都或与 Kevin Bacon 同拍过电影，或跟与 Kevin Bacon 同拍过电影的人同拍过电影。

28. 用量词和语句表达下列命题的否定。

a) 班上每个学生都喜欢数学。

b) 班上有个学生没见过计算机。

c) 班上有个学生选修过本校开设的每门数学课。

d) 班上有个学生去过校园每座楼的至少一个房间。

29. 用量词表达实数乘法的结合律。

30. 用量词表达实数乘法和加法的分配律。

练习 31～34 是根据 Lewis Carroll 的《*Symbolic Logic*》一书中的问题编写的。

31. 令$P(x)$、$Q(x)$和 $R(x)$分别为语句“x 是教授”、“x 无知”和“x 爱虚荣”。用量词、逻辑联结符和$P(x)$、$Q(x)$、$R(x)$表达下列语句。假定论域是所有人的集合。

a) 没有无知的教授。

b) 所有无知者均爱虚荣。

c) 没有爱虚荣的教授。

d) 能从 a) 和 b) 推出 c) 吗？若不能，有没有一个正确的结论？

32. 令$P(x)$、$Q(x)$和 $R(x)$分别表示语句“x 是个清楚的解释”、“x 令人满意”和“x 是借口”。假定 x 的论域是所有英语文章。用量词、逻辑联结符和$P(x)$、$Q(x)$、$R(x)$表达下列语句。

a) 所有清楚的解释都令人满意。

b) 有些借口令人不满意。

c) 有些借口不是清楚的解释。

*d) 能从 a) 和 b) 推出 c) 吗？若不能，有没有一个正确的结论？

33. 令$P(x)$、$Q(x)$、$R(x)$和 $S(x)$分别为语句“x 是婴儿”、“x 的行为符合逻辑”、“x 能管理鳄鱼”和“x 被人轻视”。假定 x 的论域是所有人的集合。用量词、逻辑联结符和$P(x)$、$Q(x)$、$R(x)$和 $S(x)$表达下列语句。

a) 婴儿行为不合逻辑。

b) 能管理鳄鱼的人不被人轻视。

c) 行为不合逻辑的人被人轻视。

d) 婴儿不能管理鳄鱼。

* e) 能从 a)、b) 和 c) 推出 d) 吗？若不能，有没有一个正确的结论？

34. 令$P(x)$、$Q(x)$、$R(x)$和 $S(x)$分别为语句“x 是只鸭子”、“x 是我的一只家禽”、“x 是一名军官”和“x 愿意跳华尔兹”。用量词、逻辑联结符和$P(x)$、$Q(x)$、$R(x)$和$S(x)$表达下列语句。

a) 没有鸭子愿意跳华尔兹。

b) 没有军官不会跳华尔兹。

c) 我所有的家禽都是鸭子。

d) 我的家禽都不是军官。

*e) 能从 a)、b) 和 c) 推出 d) 吗？若不能，有没有一个正确的结论？

35. 证明语句$\neg\exists x\,\forall y\,P(x,y)$和$\forall x\,\exists y\neg P(x,y)$有同样的真值。

36. 证明语句$\forall x(P(x)\land Q(x))$和$\forall xP(x)\land\forall xQ(x)$有同样的真值。

37. 证明$\exists x(P(x)\lor Q(x))$和$\exists xP(x)\lor\exists xQ(x)$有同样的真值。

38. 证明下列逻辑等价关系，其中 A 是不含量词的命题。

a) $(\forall xP(x))\lor A\Leftrightarrow\forall x(P(x)\lor A)$

b) $(\exists xP(x))\lor A\Leftrightarrow\exists x(P(x)\lor A)$

39. 证明下列逻辑等价关系，其中 A 是不含量词的命题。

a) $(\forall xP(x))\land A\Leftrightarrow\forall x(P(x)\land A)$

b) $(\exists xP(x))\land A\Leftrightarrow\exists x(P(x)\land A)$

40. 证明$\forall xP(x)\lor\forall xQ(x)$和$\forall x(P(x)\lor Q(x))$不是逻辑等价的。

41. 证明$\exists xP(x)\land\exists xQ(x)$和$\exists x(P(x)\land Q(x))$不是逻辑等价的。

*42. 证明$\forall xP(x)\lor\forall xQ(x)$和$\forall x\,\forall y(P(x)\lor Q(y))$逻辑等价。（新变量 y 用来正确地把量词组合在一起。）

*43. a) 证明$\forall xP(x)\land\exists xQ(x)$和$\forall x\,\exists y(P(x)\land Q(y))$逻辑等价。

b) 证明$\forall xP(x)\lor\exists xQ(x)$和$\forall x\,\exists y(P(x)\lor Q(y))$逻辑等价。

44. 符号$\exists!xP(x)$表示命题

“有唯一的 x 使$P(x)$成真。”

论域是整数集合，下列各项的真值是什么？

a) $\exists!x(x>1)$ b) $\exists!x(x^2=1)$

c) $\exists!x(x+3=2x)$ d) $\exists!x(x=x+1)$

45. 下列语句的真值是什么？

a) $\exists!xP(x)\to\exists xP(x)$

b) $\forall xP(x)\to\exists!xP(x)$

c) $\exists!x\neg P(x)\to\neg\forall xP(x)$

46. 假定论域是 1、2 和 3，用否定、合取和析取写出量化语句$\exists!xP(x)$。

*47. 用全称量词、存在量词和逻辑联结符表达量化表达式$\exists!xP(x)$。

一个语句称为前束范式（*PNF*）的充分必要条件是其表达形式为

$$Q_1x_1Q_2x_2\cdots Q_kx_kP(x_1,x_2,\cdots,x_k)$$

其中每个 Q_i，$i=1,2,\cdots,k$，或是全称量词，或是存在量词；而$P(x_1,x_2,\cdots,x_k)$是不含量词的谓词。例如$\exists x\,\forall y(P(x,y)\land Q(y))$是前束范式，而$\exists xP(x)\lor\forall xQ(x)$不是（因为并不是所有量词都先出现）。

将命题变量、谓词、T 和 F 用逻辑联结符和量词组合在一起形成的每一个语句都等价于一个前束范式。练习 49 要求的就是对这一事实的证明。

*48. 把下列语句改为前束范式。(提示：利用1.2节表1-12和表1-13的等价关系，本节表1-15以及本节的练习36～39和42～43。)

a) $\exists x P(x) \vee \exists x Q(x) \vee A$，其中 A 是不含量词的命题。

b) $\neg(\forall x P(x) \vee \forall x Q(x))$

c) $\exists x P(x) \rightarrow \exists x Q(x)$

**49. 给出并证明把任意语句变换为等价的前束范式的方法。

如果 x 大于或等于 S 中的每个数，实数 x 称为实数集合 S 的上界。如果 x 是 S 的上界，而且小于或等于 S 的所有上界，实数 x 称为实数集合 S 的最小上界。如果集合 S 有最小上界，这最小上界一定是唯一的。

50. a) 用量词表达 x 是 S 的上界这一事实。

b) 用量词表达 x 是 S 的最小上界这一事实。

51. (需要微积分知识) 用量词表达 $\lim_{x \to a} f(x)$ 不存在这一事实。

语句 $\lim_{n \to \infty} a_n = L$ 表示对每个正实数 ε，有个正整数 N，使得只要 $n > N$，就有 $|a_n - L| < \varepsilon$。

52. (需要微积分知识) 用量词表示语句 $\lim_{n \to \infty} a_n = L$。

53. (需要微积分知识) 用量词表示语句 $\lim_{n \to \infty} a_n$ 不存在。

54. (需要微积分知识) 用量词表达下面的定义：如果对每个实数 $\varepsilon > 0$，存在正整数 N，使得每一对正整数 m 和 n，只要 $m > N$ 和 $n > N$，就有 $|a_m - a_n| < \varepsilon$，则序列 $\{a_n\}$ 称为哥西序列。

55. (需要微积分知识) 用量词和逻辑联结符表达下面的定义：如果对每个实数 $\varepsilon > 0$，有无穷多个 n 使 $a_n > L - \varepsilon$ 成立，而只有有限多个 n 使 $a_n > L + \varepsilon$ 成立，则实数 L 称为序列 $\{a_n\}$ 的最大限。

1.4 集合

1.4.1 引言

我们将在本书中学习各种各样的离散结构，其中包括由元素的有序偶组成的关系，称为组合的无序元素组，以及由顶点集合和连接顶点的边集合构成的图。不仅如此，我们还将说明这些离散结构以及其他离散结构怎样用于模拟和解题。特别要讨论离散结构用于数据存储、通信和数据处理的许多例子。本节将学习的集合是构造所有其他离散结构的基础。

集合用于把对象组织在一起。通常一个集合中的对象都有相似的性质。你们学校目前在册的所有学生组成一个集合。同样，所有学校正选修一门离散数学课的学生组成一个集合。此外，你们学校在册学生中正选修一门离散数学课的所有学生组成一个集合。这个集合可以从上述两个集合中取共有的元素得到。集合语言是以构造的方式研究这种集合的工具。



注意，我们并没有说明对象是什么就已经使用这个术语了。以对象的直观概念为基础

把集合描述为一组对象，这是德国数学家康托[⊖]1895 年首先给出的。从集合的这一直观定义得到的理论导至出悖论，即逻辑上的不一致性。这是英国哲学家罗素[⊖]1902 年证明的（参见练习 26，其中描述了一个悖论）。以称为公理的基本假设为起点建立集合理论可以避免逻辑上的不一致。我们将不展开集合理论的公理描述，而是使用康托的称为朴素集合理论的原始描述，这是因为本书中考虑的所有集合均可用康托的原始理论处理而无不一致。

现在开始讨论集合。

定义 1 集合中的对象也称为该集合的元素，或成员。也说集合包含它的元素。

有几种方式描述集合，一种方式是在可能的情况下一一列出集合中的元素。我们采用在花括号之间列出所有元素的方法。例如，$\{a, b, c, d\}$ 表示含 4 个元素 a、b、c 和 d 的集合。

例 1 英语字母中所有元音字母的集合 V 可以表示为 $V = \{a, e, i, o, u\}$。 ■

例 2 小于 10 的正奇数集合 O 可以表示为 $O = \{1, 3, 5, 7, 9\}$。 ■

例 3 尽管集合常用来表示一组具有共同性质的元素，但有的集合中的元素表面上看起来毫不相干。例如，集合 $\{a, 2, \text{Fred}, \text{New Jerseg}\}$ 包含了 4 个元素：a、2、Fred 和 New Jerseg。 ■

通常用大写字母表示集合。黑体字母 **N**，**Z** 和 **R** 分别专用于表示自然数集合 $\{0, 1, 2, 3, \cdots\}$，整数集合 $\{\cdots, -2, -1, 0, 1, 2, \cdots\}$ 和实数集合。有时也用符号 $\mathbf{Z}^+$ 表示正整数集合（有人不认为 0 是自然数，所以在你阅读其他书籍时要细心检查术语自然数的用法）。

有时用花括号表示集合但并不列出它的所有元素。先列出集合中的某些元素，然后当元素的一般形式很明显时就用省略号（…）表示。

⊖ 康托(Georg Cantor，1845—1918)出生于俄罗斯的圣彼德堡，他父亲是那里一名成功的商人。康托十多岁时对数学产生了浓厚的兴趣。1862 年他在苏黎世开始了他的大学学习，不过在他父亲去世时就离开了那里。1863 年他在柏林大学继续学习，并得到著名数学家 Weierstrass、Kummer 和 Kronecker 的指导。1867 年在完成了一篇数论方面的博士论文后他获得博士学位。1869 年康托得到哈雷大学的一个职位，并在那里一直工作到去世。

康托是公认的集合论奠基人。他在这一领域的贡献包括实数集合不可数性的发现。他在数学分析方面的贡献也引人注目。康托对哲学也有兴趣，并写了若干论文，将他在集合理论上的工作与形而上学联系在一起。

康托 1874 年结婚，有 5 个子女。他忧郁的气质与妻子的乐观性情相互平衡。尽管他从父亲那里得到大笔遗产，但作为教授的收入却很少。为此他曾试图得到柏林大学一个待遇更高的位置。对他的这一任命被 Kronecker 阻止了，因为 Kronecker 不同意康托集合论的观点。康托晚年受到精神疾病的折磨，1918 年死于精神病诊所。

⊖ 罗素(Bertrand Russell，1872—1970)罗素生于一个以积极参与进步运动、热烈地投身自由事业而著名的英格兰家庭。年幼时就成为孤儿的罗素由祖父母扶养，并在家里接受教育。1890 年他进入剑桥的 Trinity 学院学习数学和伦理学。他在几何学基础方面的工作为他赢得了一个研究员职位。1910 年 Trinity 学院任命他教授逻辑和数学原理的课程。

罗素毕生为进步事业而战斗。他有强烈的和平主义观点，他对第一次世界大战的抗议使他失去了 Trinity 学院的位置。由于一篇被认为具有煽动性的文章使他在 1918 年被囚禁 6 个月。罗素还为英国妇女的选举权而斗争。1961 年在他 89 岁高龄时第二次入狱，原因是加入呼吁核裁军的抗议活动。

罗素最伟大的工作是他提出的可以作为所有数学学科基础的原理。他最著名的文章是与怀特海德（Alfred North Whitehead）合写的《*Principia Mathematica*》。这篇文章试图用一组基本公理推导出所有数学。他还撰写了许多书籍，内容包括哲学、物理和他的政治观点。1950 年罗素赢得诺贝尔文学奖。

例4 小于100的正整数集合可以表示为 $\{1, 2, 3, \cdots, 99\}$。 ■

由于许多数学语句断定两组以不同方式描述的对象实际上是同一个集合，我们需要理解两个集合相等的含义。

定义2 *两个集合相等当且仅当它们有同样的元素。*

例5 集合 $\{1, 3, 5\}$ 和 $\{3, 5, 1\}$ 相等，因为它们有同样的元素。注意列出元素的顺序不起作用，还要注意同一个元素被列出来不止一次也没关系。所以 $\{1, 3, 3, 3, 5, 5, 5, 5\}$ 和 $\{1, 3, 5\}$ 是同一个集合，因为它们有同样的元素。 ■

描述集合的另一方式是使用*集合构造符号*。我们给出作为集合的成员必须具有的性质，以此来刻画集合的所有元素。例如，小于10的所有奇数的集合 O 可以写成

$O=\{x \mid x$ 是小于10的奇数$\}$

在无法给出集合所有元素时常用这一类方法描述集合。例如，所有实数的集合可以写成

$\mathbf{R}=\{x \mid x$ 为实数$\}$

还可以用文氏图形象地表示集合。文氏图是以英国数学家John Venn[⊖]的名字命名的，他在1881年介绍了这种图的使用。我们所考虑的所有对象的集合 U，称为全集。在文氏图中全集用长方形表示。在长方形内部，圆或其他几何图形用于表示集合，有时用点来表示集合中特定的元素。文氏图常用于表示集合之间的关系。我们用下面的例子解释怎样使用文氏图。

例6 画一个表示英语字母中元音字母集合 V 的文氏图

解 画一个长方形表示全集 U，也就是26个英文字母的集合。在长方形中画一个圆表示集合 V，在圆中用点表示集合 V 的元素（见图1-1）。 ■

我们现在介绍描述集合成员关系的记号。我们用 $a \in A$ 表示 a 是集合 A 的一个元素。记号 $a \notin A$ 表示 a 不是集合 A 的成员。注意小写字母通常用于表示集合元素。

有一个不含任何元素的特殊集合，称为空集，用 $\emptyset$ 表示。空集也可以用 $\{\ \}$ 表示（这是用花括号内列出所有元素的方式表示集合）。经常具有一定性质的元素组成的集合其实就是空集。例如，大于自身的平方的所有正整数的集合是空集。

定义3 *集合 A 是集合 B 的子集当且仅当 A 的每个元素也是 B 的元素。我们用记号 $A \subseteq B$ 表示 A 是 B 的子集。*

⊖ 文氏（John Venn，1834—1923）出生于伦敦郊区的一个慈善家庭。他在伦敦上学，并于1857年获得剑桥Caius学院的数学学位。他被选为这一学院的研究员并任此职直至去世。1859年他接受牧师的圣职，但在短短的宗教工作以后回到了剑桥，并在那里创建了伦理学教育。除数学方面的工作之外，文氏还对历史有兴趣，他写了大量关于他的学院和家庭的作品。

文氏所著《*Symbolic Logic*》一书澄清了最初由布尔引入的若干思想。在这本书中文氏提出了一种系统的研究方法，其中使用了后人称为文氏图的几何图形。今天这些图形主要用于分析逻辑论证及说明集合之间的关系。文氏的成绩不仅是符号逻辑方面的工作，还包括他对概率论的贡献，这些贡献在他编写的广为采用的概率论教科书中都能找到。

我们看到，$A \subseteq B$ 当且仅当量化语句

$$\forall x(x \in A \to x \in B)$$

为真。例如，小于 10 的所有奇数的集合是小于 10 的所有正整数的集合的子集。你们学校主修计算机科学的学生的集合是你们学校全体学生集合的子集。

空集是每个集合的子集，也就是说，

$$\emptyset \subseteq S$$

其中 S 是个集合。要证明空集是 S 的子集，必须证明空集的每个元素也在 S 中。换言之，必须证明蕴含关系“如果 $x \in \emptyset$，那么 $x \in S$”永远成立。只需注意到此蕴含关系的前提（即 $x \in \emptyset$）永不成立，就知道这蕴含关系总是为真。于是空集是每个集合的子集。此外，每个集合都是它自己的子集（读者应证明这一关系）。所以如果 P 是个集合，则有 $\emptyset \subseteq P$ 及 $P \subseteq P$。

如果我们要强调集合 A 是集合 B 的子集，但 $A \neq B$，就写成 $A \subset B$，并说 A 是 B 的真子集。文氏图可以用来表示集合 A 是集合 B 的子集。我们把全集 U 画成长方形。在这长方形中画一表示 B 的圆。由于 A 是 B 的子集，我们在代表 B 的圆内画表示 A 的圆。图 1-2 所示就是这一关系。

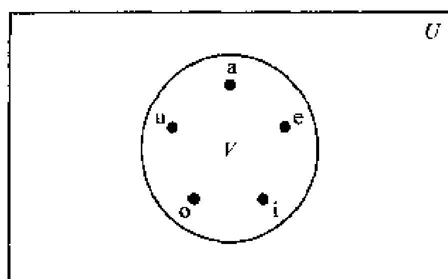


图 1-1 元音字母集合的文氏图

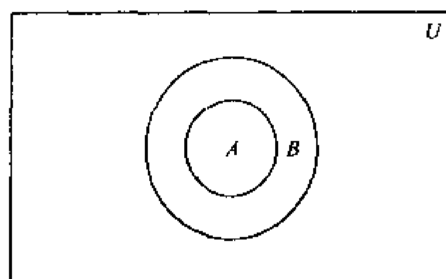


图 1-2 表示 A 是 B 的子集的文氏图

证明两个集合具有相同元素的一个方法是证明它们互为另一个的子集。换言之，可以证明，如果 A 和 B 均为集合，且 $A \subseteq B$ 和 $B \subseteq A$，则 $A = B$。实践表明，这是证明集合相等的一种有用的方式。

集合可以以其他集合作为它的成员。例如，下面列出的就是这样的集合：

$$\{\emptyset, \{a\}, \{b\}, \{a, b\}\} \text{和} \{x \mid x \text{是集合}\{a, b\}\text{的子集}\}$$

注意，这两个集合相等。

集合广泛用于计数问题。这类问题需要讨论集合的大小。

定义 4 令 S 为集合。若 S 中恰有 n 个不同的元素，n 是非负整数，就说 S 是*有限集合*，而 n 是 S 的*基数*。S 的基数用 $|S|$ 表示。

例 7 令 A 为小于 10 的正奇数集合，则 $|A| = 5$。 ■

例 8 令 S 为英语字母集，那么 $|S| = 26$。 ■

例 9 由于空集没有元素，所以 $|\emptyset| = 0$。 ■

我们对不是有限的集合也有兴趣。

定义 5 如果一个集合不是有限的，就说它是*无限的*。

例 10 正整数集合是无限的。 ■

无限集合的基数将在 1.7 节讨论。在那一节将讨论可数集合的含义，并证明有些集合是可数的，有些不是可数的。

1.4.2 幂集合

许多问题都要检查一个集合的元素所有可能的组合，看它们是否具有某种性质。为了考虑集合元素所有可能的组合，我们构造一个新集合，它以 S 的所有子集作为它的元素。

定义 6 已知集合 S，S 的*幂集合*是集合 S 所有子集的集合。S 的幂集合用 $P(S)$ 表示。

例 11 集合 $\{0,1,2\}$ 的幂集合是什么？

解 幂集合 $P(\{0,1,2\})$ 是 $\{0,1,2\}$ 所有子集的集合。因此

$$P(\{0,1,2\}) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$$

注意，空集和 $\{0,1,2\}$ 自身都是这个子集集合的成员。 ■

例 12 空集的幂集合是什么？集合 $\{\emptyset\}$ 的幂集合是什么？

解 空集只有一个子集，这就是它自己。因此

$$P(\emptyset) = \{\emptyset\}$$

集合 $\{\emptyset\}$ 有两个子集，即 $\emptyset$ 和集合 $\{\emptyset\}$ 自身。于是

$$P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$$ ■

如果一个集合有 n 个元素，那么它的幂集合有 2^n 个元素。我们将在本书以后的几节中以不同的方式证明这一事实。

1.4.3 笛卡儿积

一组元素中元素的次序往往是重要的。由于集合是无序的，必须用不同的结构来表示有序的一组元素，这个结构由*有序 n 元组*来提供。

定义 7 *有序 n 元组* $(a_1, a_2, \cdots, a_n)$ 是以 a_1 为第 1 个元素，a_2 为第 2 个元素，$\cdots$，a_n 为第 n 个元素的有序组。

只有在两个有序 n 元组每一对对应的元素都相等时，它们才相等。换言之，$(a_1, a_2, \cdots, a_n) = (b_1, b_2, \cdots, b_n)$ 当且仅当对于 $i = 1, 2, \cdots, n$，$a_i = b_i$。2 元组特称为*有序偶*。有序偶 (a, b) 和 (c, d) 相等当且仅当 $a = c$ 和 $b = d$。注意，除非 $a = b$，否则 (a, b) 和 (b, a) 不相等。

在随后几章中将要学习的许多离散结构均以 René Descartes㊀命名的集合的笛卡儿积的概念为基础。我们先定义两个集合的笛卡儿积。

定义 8　令 A 和 B 为集合。A 和 B 的笛卡儿积用 $A\times B$ 表示，是所有有序偶（a，b）的集合，其中 $a\in A$ 而 $b\in B$。于是

$$A\times B=\{(a,b)\mid a\in A\wedge b\in B\}$$

例 13　令 A 为某大学所有学生的集合，B 表示该大学开设的所有课程的集合。A 和 B 的笛卡儿积 $A\times B$ 是什么？

解　笛卡儿积 $A\times B$ 由形为 (a,b) 的所有有序偶组成，其中 a 是个学生，而 b 是该校开的一门课。集合 $A\times B$ 可以用来表示该校学生选课的所有可能情况。■

例 14　什么是 $A=\{1,2\}$ 和 $B=\{a,b,c\}$ 的笛卡儿积？

解　笛卡儿积 $A\times B$ 是

$$A\times B=\{(1,a),(1,b),(1,c),(2,a),(2,b),(2,c)\}$$ ■

除非 $A=\emptyset$ 或 $B=\emptyset$（这样 $A\times B=\emptyset$）或除非 $A=B$，否则 $A\times B$ 和 $B\times A$ 不相等（参看本节末尾练习 24）。这在下例中说明。

例 15　证明笛卡儿积 $B\times A$ 和笛卡儿积 $A\times B$ 不相等，其中 A 和 B 为例 14 中的集合。

解　笛卡儿积 $B\times A$ 是

$$B\times A=\{(a,1),(a,2),(b,1),(b,2),(c,1),(c,2)\}$$

这不等于例 14 中得到的 $A\times B$。■

对于两个以上的集合也可以定义笛卡儿积。

定义 9　集合 A_1，A_2，…，A_n 的笛卡儿集用 $A_1\times A_2\times\cdots\times A_n$ 表示，这是有序 n 元组（a_1，a_2，…，a_n）的集合，其中对于 $i=1$，2，…，n，$a_i\in A_i$。换言之，

$$A_1\times A_2\times\cdots\times A_n=\{(a_1,a_2,\cdots,a_n)\mid a_i\in A_i,i=1,2,\cdots,n\}$$

例 16　什么是笛卡儿积 $A\times B\times C$，其中 $A=\{0,1\},B=\{1,2\},C=\{0,1,2\}$？

㊀ 笛卡儿(René Descartes,1596—1650)出生于法国 Tours 的一个贵族家庭，Tours 位于巴黎西南约 200 英里处。他是他父亲第一位妻子的第三个孩子，在他出生之后母亲就去世了。由于笛卡儿健康欠佳，他父亲，一位省里的法官，没有让他接受正规教育，直到 8 岁他才进入 La Flèche 的 Jesuit 学院。这所学校的校长喜欢他，并因他身体虚弱而允许他晚起床。从那时起笛卡儿即把早晨时间花在床上，他自己认为这是他思考最富成果的时间。

1612 年笛卡儿离开学校去了巴黎，并在那里学了两年数学。1616 年他获得 Poitiers 大学的法律学位。18 岁时笛卡儿厌倦了学习，决定看看外部世界。他回到巴黎并成为一名成功的赌徒。不过他逐渐讨厌这种不洁的生活，搬到了 Saint-Germain 城的郊区，一心专注于数学学习。当他的赌徒朋友找到他以后，他决心离开法国并参了军。不过他从未参加过战斗。一天，当他为躲避寒冷而呆在军营一间过热的房间里时，他做了几个热昏了的梦，梦中显示他将以做数学家和哲学家为职业。

结束军旅生涯以后，他游遍欧洲。然后在巴黎过了几年，学习数学和哲学，并制造了几个光学仪器。笛卡儿决定去荷兰，并在那里四处漫游 20 年，完成了他最重要的工作。这期间他写了几本书，包括《Discours》，该书中有他对解析几何的贡献，这是他最出名的工作。笛卡儿对哲学也有基础性的贡献。

1649 年笛卡儿受克里斯蒂娜女王邀请访问瑞典宫廷，并做她的哲学老师。尽管他不情愿生活在他所说的“岩石和冰雪之间的狗熊的乐土”，但最终还是接受邀请，迁到了瑞典。不幸的是，1649—1650 年的冬季分外的寒冷，笛卡儿染上了肺炎，于二月中旬去世。

解 这一笛卡儿积由所有有序三元组(a,b,c)组成，其中$a\in A$，$b\in B$，$c\in C$。因此，

$$A\times B\times C=\{(0,1,0),(0,1,1),(0,1,2),(0,2,0),(0,2,1),(0,2,2),(1,1,0),(1,1,1),(1,1,2),(1,2,0),(1,2,1),(1,2,2)\}$$ ■

练习

1. 列出下述集合的成员。
   a) $\{x\mid x$ 是使得 $x^2=1$ 的实数$\}$
   b) $\{x\mid x$ 是小于 12 的正整数$\}$
   c) $\{x\mid x$ 是某个整数的平方且 $x<100\}$
   d) $\{x\mid x$ 是整数且 $x^2=2\}$
2. 用集合构造符号，给出下列每个集合的描述。
   a) $\{0, 3, 6, 9, 12\}$
   b) $\{-3,-2,-1,0,1,2,3\}$
   c) $\{m, n, o, p\}$
3. 判断下面每对集合是否相等。
   a) $\{1, 3, 3, 3, 5, 5, 5, 5, 5\}$，$\{5, 3, 1\}$
   b) $\{\{1\}\}$，$\{1,\{1\}\}$
   c) $\emptyset$，$\{\emptyset\}$
4. 设$A=\{2,4,6\}$，$B=\{2,6\}$，$C=\{4,6\}$，$D=\{4,6,8\}$。判断这些集合中哪个是另外一个的子集。
5. 对下面的每个集合，判断 2 是否为它的元素。
   a) $\{x\in\mathbf{R}\mid x$ 是大于 1 的整数$\}$
   b) $\{x\in\mathbf{R}\mid x$ 是某整数的平方$\}$
   c) $\{2, \{2\}\}$
   d) $\{\{2\},\{\{2\}\}\}$
   e) $\{\{2\},\{2,\{2\}\}\}$
   f) $\{\{\{2\}\}\}$
6. 对练习 5 中的每个集合，判断 $\{2\}$ 是否它的一个元素。
7. 判断下列语句是真还是假。
   a) $x\in\{x\}$
   b) $\{x\}\subseteq\{x\}$
   c) $\{x\}\in\{x\}$
   d) $\{x\}\in\{\{x\}\}$
   e) $\emptyset\subseteq\{x\}$
   f) $\emptyset\in\{x\}$
8. 用文氏图说明集合关系 $A\subseteq B$ 和 $B\subseteq C$。
9. 假定 A、B 和 C 为集合，且 $A\subseteq B$，$B\subseteq C$。证明 $A\subseteq C$。
10. 找出两个集合 A 和 B，使得 $A\in B$ 且 $A\subseteq B$。

11. 下列各集合的基数是什么？

a) $\{a\}$

b) $\{\{a\}\}$

c) $\{a,\{a\}\}$

d) $\{a,\{a\},\{a,\{a\}\}\}$

12. 下列各集合的基数是什么？

a) $\emptyset$

b) $\{\emptyset\}$

c) $\{\emptyset,\{\emptyset\}\}$

d) $\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}$

13. 找出下列各集合的幂集合。

a) $\{a\}$

b) $\{a,b\}$

c) $\{\emptyset,\{\emptyset\}\}$

14. 如果 A 和 B 是两个集合，且有相同的幂集合，能否肯定 $A=B$？

15. 下列各集合各有多少个元素？

a) $P(\{a,b,\{a,b\}\})$

b) $P(\{\emptyset,a,\{a\},\{\{a\}\}\})$

c) $P(P(\emptyset))$

16. 判断下列各集合是否为某集合的幂集合。

a) $\emptyset$

b) $\{\emptyset,\{a\}\}$

c) $\{\emptyset,\{a\},\{\emptyset,a\}\}$

d) $\{\emptyset,\{a\},\{b\},\{a,b\}\}$

17. 令 $A=\{a,b,c,d\}$，$B=\{y,z\}$。求

a) $A\times B$

b) $B\times A$

18. 令 A 为某大学的数学系开设课程的集合，B 为该大学所有数学教授的集合，笛卡儿集 $A\times B$ 是什么？

19. 什么是笛卡儿集 $A\times B\times C$，其中 A 是所有航线的集合，B 和 C 都是所有美国城市的集合。

20. 假定 $A\times B=\emptyset$，其中 A 和 B 为集合，由此你得出什么结论？

21. 令 A 为集合，求证 $\emptyset\times A=A\times\emptyset=\emptyset$。

22. 令 $A=\{a,b,c\}$，$B=\{x,y\}$，$C=\{0,1\}$。求

a) $A\times B\times C$

b) $C\times B\times A$

c) $C\times A\times B$

d) $B\times B\times B$

23. 如果 A 有 m 个元素，B 有 n 个元素，$A\times B$ 有多少个元素？

24. 求证除非 $A=B$，否则 $A\times B\neq B\times A$，其中 A 和 B 均为非空集合。

*25. 证明序偶 (a,b) 可以用集合术语定义为 $\{\{a\},\{a,b\}\}$。[提示：首先证明 $\{\{a\},\{a,b\}\}=\{\{c\},\{c,d\}\}$ 当且仅当 $a=c$ 且 $b=d$。]

*26. 本练习介绍罗素悖论。令集合 S 为包含所有不以自身为元素的那些集合，即 $S=\{x\mid x\notin x\}$。

a) 证明从 S 是它自己的一个元素的假设能推出矛盾。

b) 证明从 S 不是它自己的一个元素的假设能推出矛盾。

从 a) 和 b) 知 S 不可能是其定义中给出的那个集合。不过这一悖论是可以避免的，只要对可以作为集合元素的类型加以限制即可。

*27. 给出可以列出有限集合所有子集的步骤。

1.5　集合运算

1.5.1　引言

两个集合可以以许多不同的方式结合在一起。例如，从学校主修数学课的集合和主修计算机科学课的集合人手，可以构成主修数学或计算机科学的学生集合，既主修数学又主修计算机科学的学生集合，不主修数学的学生集合，等等。

定义 1　令 A 和 B 为集合。A 和 B 的并集用 $A\cup B$ 表示，这是 A 或 B 中或同时在 A 和 B 中的元素组成的集合。

元素 x 属于 A 和 B 的并集当且仅当 x 属于 A 或 x 属于 B。这说明

$$A\cup B=\{x\mid x\in A\vee x\in B\}$$

图 1-3 中的文氏图表示集合 A 和 B 的并集。代表集合 A 的圆圈内和代表 B 的圆圈内的阴影区域表示 A 和 B 的并集。

我们将给出集合并集的例子。

例 1　集合 $\{1,3,5\}$ 和集合 $\{1,2,3\}$ 的并集是集合 $\{1,2,3,5\}$，即

$$\{1,3,5\}\cup\{1,2,3\}=\{1,2,3,5\}$$ ■

例 2　学校主修计算机科学的学生集合与主修数学的学生集合的并集，是或主修数学、或主修计算机科学或同时主修这两门课的学生的集合。■

定义 2　令 A 和 B 的集合。A 和 B 的交集用 $A\cap B$ 表示，这是既在 A 中又在 B 中的那些元素的集合。

元素 x 属于集合 A 和 B 的交集当且仅当 x 属于 A 而且 x 属于 B。这说明

$$A\cap B=\{x\mid x\in A\wedge x\in B\}$$

图 1-4 中的文氏图表示集合 A 和 B 的交集。同时在代表 A 和 B 的两个园之内的阴影区域表示 A 和 B 的交集。

我们给出交集的几个例子。

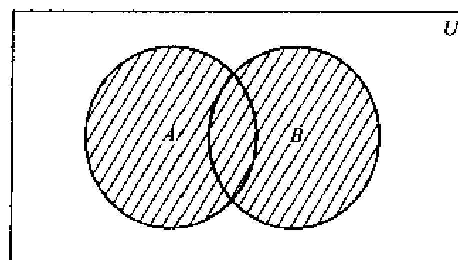


图 1-3　表示 A 和 B 并集的文氏图

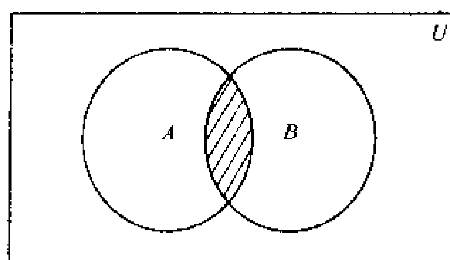


图 1-4　表示 A 和 B 交集的文氏图

例 3　集合$\{1,3,5\}$和$\{1,2,3\}$的交集是$\{1,3\}$，即$\{1,3,5\}\cap\{1,2,3\}=\{1,3\}$。■

例 4　学校所有主修计算机科学的学生集合与所有主修数学的学生集合，是所有既主修计算机科学又主修数学的学生的集合。■

定义 3　如果两个集合的交集为空集，就说它们不相交。

例 5　令 $A=\{1,3,5,7,9\}$ 而 $B=\{2,4,6,8,10\}$。由于 $A\cap B=\emptyset$，A 和 B 不相交。■

我们往往有兴趣求出集合的并集的基数。要计算两个有限集合 A 和 B 的并集的元素个数，须注意$|A|+|B|$把只属于 A 或只属于 B 的元素数了一次，而既属于 A 又属于 B 的元素数了两次。因此，如果从$|A|+|B|$中减去既属于 A 又属于 B 的元素的个数，则 $A\cap B$ 中的元素也就只数一次。于是

$$|A\cup B|=|A|+|B|-|A\cap B|$$

把这一结果推广到任意多个集合的并集，就得到所谓的*包含排斥原理*（或简称容斥原理）。容斥原理是用于枚举的一项重要技术。我们将在第 4 章和第 5 章详细讨论这一原理和其他的计数技术。

还有其他组合集合的重要方式。

定义 4　令 A 和 B 为集合。A 和 B 的差集用 $A-B$ 表示，这是包含只属于 A 而不属于 B 的所有元素的集合。A 和 B 的差集也称为 B 对于 A 的补集。

元素 x 属于 A 和 B 的差集当且仅当 $x\in A$ 且 $x\notin B$，这说明

$$A-B=\{x\mid x\in A\wedge x\notin B\}$$

图 1-5 中的文氏图表示集合 A 和 B 的差集。既在表示集合 A 的圆圈内部也在表示集合 B 的圆圈外部的阴影区域表示 $A-B$。

让我们举几个差集的例子。

例 6　集合 $\{1,3,5\}$ 和 $\{1,2,3\}$ 的差集是 $\{5\}$；即$\{1,3,5\}-\{1,2,3\}=\{5\}$。这不同于 $\{1,2,3\}$ 和 $\{1,3,5\}$ 的差集 $\{2\}$。■

例 7　学校主修计算机科学的学生集合和主修数学的学生集合的差集是学校主修计算机科学但不主修数学的学生集合。■

一旦指定了全集 U，就可以定义集合的补集。

定义 5　令 U 为全集。集合 A 的补集用 $\overline{A}$ 表示，这是 A 对于 U 的补集。集合 A 的补集是 $U-A$。

元素 x 属于 $\overline{A}$ 当且仅当 $x\notin A$。这说明

$$\overline{A}=\{x \mid x\notin A\}。$$

图 6 中代表集合 A 的园圈外面的阴影区域表示 $\overline{A}$。

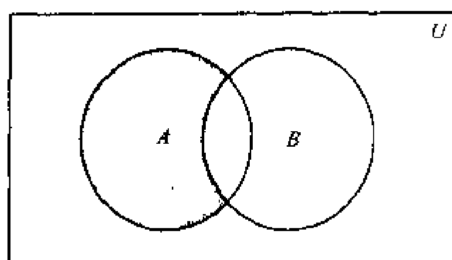


图 1-5　A 和 B 的差集的文氏图

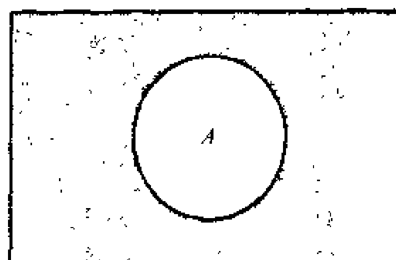


图 1-6　集合 A 的补集的文氏图

我们举几个补集的例子。

例 8　令 $A=\{a,e,i,o,u\}$（全集为英语字母集）。那么

$$\overline{A}=\{b,c,d,f,g,h,j,k,l,m,n,p,q,r,s,t,v,w,x,y,z\}$$ ■

例 9　令 A 为大于 10 的正整数的集合（全集为正整数集合）。那么

$$\overline{A}=\{1,2,3,4,5,6,7,8,9,10\}$$ ■

1.5.2　集合相等

表 1-17 列出了最重要的集合等式。本节将用三种不同的方法证明其中的几个。介绍这些方法是想说明，对一个问题的解往往有不同的途径。表中未证明的等式留给读者作练习。读者应该注意这些集合等式和 1.2 节讨论的逻辑等价的相似之处。事实上，这里给出的集合等式可以直接由对应的逻辑等价证明。不仅如此，而且这两者都是布尔代数（在第 9 章讨论）中成立的恒等式的特殊情况。

证明集合相等的一种方法是证明两者中任何一个都是另一个的子集。我们将以确立德摩根第二定律为例说明这一方法。

例 10　用证明 $\overline{A\cap B}$ 和 $\overline{A}\cup\overline{B}$ 互为子集的方法证明 $\overline{A\cap B}=\overline{A}\cup\overline{B}$。

解　首先假定 $x\in\overline{A\cap B}$。于是 $x\notin A\cap B$。这表示 $x\notin A$ 或 $x\notin B$。所以 $x\in\overline{A}$ 或 $x\in\overline{B}$。于是 $x\in\overline{A}\cup\overline{B}$。这表明 $\overline{A\cap B}\subseteq\overline{A}\cup\overline{B}$。

现在假定 $x\in\overline{A}\cup\overline{B}$。那么 $x\in\overline{A}$ 或 $x\in\overline{B}$。于是 $x\notin A$ 或 $x\notin B$，从而 $x\notin A\cap B$。这表明 $\overline{A}\cup\overline{B}\subseteq\overline{A\cap B}$ 由于已经证明了这两个集合互为子集，它们必定相等，等式得证。 ■

另一个证明集合相等的方法是使用集和构造符和逻辑规则。考虑德摩根第二定律的下述证明。

表 1-17 集合相等

等式	名称
$A\cup\emptyset=A$ $A\cap U=A$	恒等律
$A\cup U=U$ $A\cap\emptyset=\emptyset$	支配律
$A\cup A=A$ $A\cap A=A$	幂等律
$\overline{(\overline{A})}=A$	补集律
$A\cup B=B\cup A$ $A\cap B=B\cap A$	交换律
$A\cup(B\cup C)=(A\cup B)\cup C$ $A\cap(B\cap C)=(A\cap B)\cap C$	结合律
$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$ $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$	分配律
$\overline{A\cup B}=\overline{A}\cap\overline{B}$ $\overline{A\cap B}=\overline{A}\cup\overline{B}$	德摩根定律

例 11 用集合构造符和逻辑等价证明$\overline{A\cap B}=\overline{A}\cup\overline{B}$。

解 下列一串等式提供了这一相等关系的证明：

$$\begin{aligned}\overline{A\cap B}&=\{x\mid x\notin A\cap B\}\\&=\{x\mid\neg(x\in A\cap B)\}\\&=\{x\mid\neg(x\in A\wedge x\in B)\}\\&=\{x\mid x\notin A\vee x\notin B\}\\&=\{x\mid x\in\overline{A}\vee x\in\overline{B}\}\\&=\{x\mid x\in\overline{A}\cap\overline{B}\}\end{aligned}$$

注意在这一串等式中的第四个等号处使用了逻辑等价的德摩根第二定律。 ■

还可以用成员表来证明集合相等。我们考虑一个元素可能属于的集合的每一种组合，并证明在同样的集合组合中的元素属于等式两边的集合。用 1 表示元素属于一个集合，用 0 表示元素不属于一个集合（读者应注意成员表和真值表的相似之处）。

例 12 用成员表证明$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$。

解 表 1-18 给出了这些集合组合的成员表。这表格有 8 行。由于对应于$A\cap(B\cup C)$和$(A\cap B)\cup(A\cap C)$的两列相同，等式有效。 ■

用我们已经证明的这些集合等式可以证明其他的集合等式。考虑下面的例子。

例 13 令 A，B，C 为集合。证明

$$\overline{A\cup(B\cap C)}=(\overline{C}\cup\overline{B})\cap\overline{A}$$

表 1-18 分配性质的成员表

A	B	C	$B\cup C$	$A\cap(B\cup C)$	$A\cap B$	$A\cap C$	$(A\cap B)\cup(A\cap C)$
1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

解 我们有

$$\begin{aligned}\overline{A\cup(B\cap C)} &= \overline{A}\cap(\overline{B\cap C}) && \text{德摩根第一律}\\ &= \overline{A}\cap(\overline{B}\cup\overline{C}) && \text{德摩根第二律}\\ &= (\overline{B}\cup\overline{C})\cap\overline{A} && \text{交集的交换律}\\ &= (\overline{C}\cup\overline{B})\cap\overline{A} && \text{并集的交换律}\end{aligned}$$



1.5.3 扩展的并集和交集

由于集合的并集和交集满足结合律，所以只要 A，B，C 为集合，$A\cup B\cup C$ 和 $A\cap B\cap C$ 均有定义。注意 $A\cup B\cup C$ 包含那些至少属于 A，B，C 中一个集合的元素，而 $A\cap B\cap C$ 包含那些属于 A，B，C 全部三个集合的元素。三个集合 A，B，C 的这两个组合如图 1-7 所示。

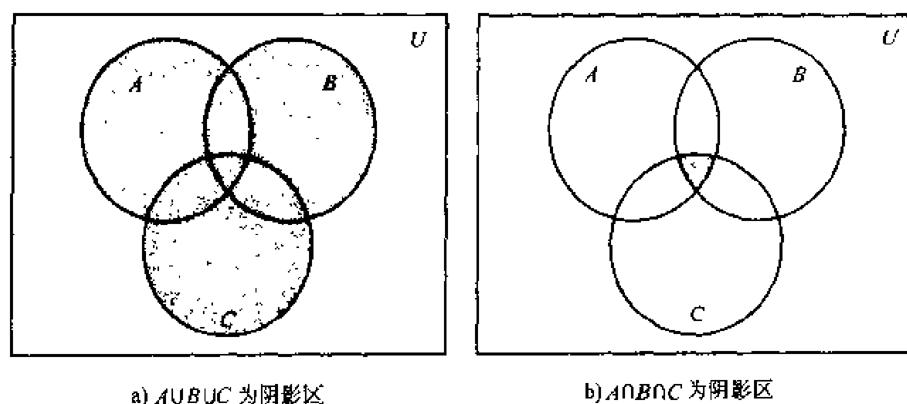


图 1-7 集合 A，B，C 的并集和交集

例 14 令 $A=\{0,2,4,6,8\}$，$B=\{0,1,2,3,4\}$，$C=\{0,3,6,9\}$。$A\cup B\cup C$ 和 $A\cap B\cap C$ 是什么集合？

解 $A\cup B\cup C$ 包括那些至少属于 A，B，C 之一的元素。所以

$$A\cup B\cup C=\{0,1,2,3,4,6,8,9\}$$

集合$A\cap B\cap C$包括那些属于全部三个集合的元素。因此

$$A\cap B\cap C=\{0\}$$

■

可以考虑任意多个集合的并集和交集。我们采用下面的定义。

定义 6 一组集合的并集是包含那些至少是这组集合中一个集合成员的元素的集合。

用记号

$$A_1\cup A_2\cup\cdots\cup A_n=\bigcup_{i=1}^{n}A_i$$

表示集合A_1，A_2，…，A_n的并集。

定义 7 一组集合的交集是包含那些属于这组集合中所有成员集合的元素的集合。

用记号

$$A_1\cap A_2\cap\cdots\cap A_n=\bigcap_{i=1}^{n}A_i$$

表示集合A_1，A_2，…，A_n的交集。我们用下面的例子说明扩展的并集和交集。

例 15 令$A_i=\{i,i+1,i+2,\cdots\}$。那么

$$\bigcup_{i=1}^{n}A_i=\bigcup_{i=1}^{n}\{i,i+1,i+2,\cdots\}=\{1,2,3,\cdots\}$$

而

$$\bigcap_{i=1}^{n}A_i=\bigcap_{i=1}^{n}\{i,i+1,i+2,\cdots\}=\{n,n+1,n+2,\cdots\}$$

■

1.5.4 集合的计算机表示

计算机表示集合的方式各种各样。一种办法是把集合的元素无序地存储起来。可是如果这样做，在做集合的并集、交集或差集等运算时会浪费时间，因为这些运算将需要大量的元素检索。我们将要介绍一种利用全集元素的一个任意排序存放元素以表示集合的方法。集合的这种表示法使我们很容易计算集合的组合。

假定全集U是有限的（而且大小合适，使U的元素个数不超过计算机能使用的内存量）。首先为U的元素任意规定一个顺序，例如a_1，a_2，…，a_n。于是可以用长度为n的位串表示U的子集A：如果a_i属于A，则位串中第i位是1；如果a_i不属于A，则位串中第i位是0。下面的例子阐明了这一方法。

例 16 令$U=\{1,2,3,4,5,6,7,8,9,10\}$，而且$U$的元素从小到大排序，即$a_i=i$。表示$U$中所有奇数的子集、所有偶数的子集和不超过5的整数的子集的位串是什么？

解 表示U中所有奇数的子集即$\{1，3，5，7，9\}$的位串，其第1、3、5、7、9位为1，其他位为0，即

10101 01010

（我们已把长度为10的位串分成长度为5的两段以便阅读，因为长位串不易读）。类似地，U中所有偶数的子集即$\{2，4，5，8，10\}$，由位串

01010　10101

表示。U 中不超过 5 的所有整数的集合即 {1，2，3，4，5}，由位串

11111　00000 ■

表示。

用位串表示集合便于计算集合的补集、并集、交集和差集。要从表示集合的位串计算它的补集的位串，只须简单地把每个 1 改为 0，每个 0 改为 1，因为 $x \in A$ 当且仅当 $x \notin \overline{A}$。注意，当把每个字位看成是真值时，上述运算对应于取每个字位的非，因为 1 表示真，0 表示假。

例 17　我们已经知道集合 {1，3，5，7，9} 的位串（全集为 {1，2，3，4，5，6，7，8，9，10}）是

10101 01010

它的补集的位串是什么？

解　用 0 取代 1，用 1 取代 0，即可得到此集合的补集的位串

01010 10101

这对应着集合{2,4,6,8,10}。 ■

要得到两个集合的并集和交集的位串，我们可以对表示这两个集合的位串按位做布尔运算。只要两个位串的第 i 字位有一个是 1，则并集的位串的第 i 位是 1，当两个字位都是 0 时为 0。因此，并集的位串是两个集合位串的按位或（bitwise *OR*）。当两个位串的第 i 字位均为 1 时，交集的位串第 i 位为 1，否则为 0。因此交集的位串是两个集合位串的按位与(bitwise *AND*)。

例 18　集合 {1，2，3，4，5} 和 {1，3，5，7，9} 的位串分别是 11111 00000 和 10101 01010。用位串找出它们的并集和交集。

解　这两个集合的并集的位串是

$11111\ 00000 \vee 10101\ 01010 = 11111\ 01010$

它表示的集合是 {1，2，3，4，5，7，9}。这两个集合的交集的位串是

$11111\ 00000 \wedge 10101\ 01010 = 10101\ 00000$

它表示的集合是 {1，3，5}。 ■

练习

1. 令 A 为住在离学校一英里以内的所有学生的集合，B 是走路上学的所有学生的集合。描述一下下列各集合中的学生：
   a) $A \cap B$　　　b) $A \cup B$
   c) $A - B$　　　d) $B - A$
2. 假定 A 是学校二年级的学生集合，B 是学校上离散数学课的学生集合。用 A 和 B 来表示下列各个集合。
   a) 学校二年级上离散数学课的学生集合。

b) 学校二年级不上离散数学课的学生集合。

c) 学校二年级的或上离散数学课的学生集合。

d) 学校既不在二年级的也不上离散数学课的学生集合。

3. 令 $A=\{1,2,3,4,5\}$，$B=\{0,3,6\}$。求

a) $A\cup B$　　b) $A\cap B$

c) $A-B$　　d) $B-A$

4. 令 $A=\{a,b,c,d,e\}$，$B=\{a,b,c,d,e,f,g,h\}$。求

a) $A\cup B$　　b) $A\cap B$

c) $A-B$　　d) $B-A$

5. 令 A 为集合。证明 $\overline{\overline{A}}=A$。

6. 令 A 为集合。证明

a) $A\cup\emptyset=A$　　b) $A\cap\emptyset=\emptyset$

c) $A\cup A=A$　　d) $A\cap A=A$

e) $A-\emptyset=A$　　f) $A\cup U=U$

g) $A\cap U=A$　　h) $\emptyset-A=\emptyset$

7. 令 A 和 B 为集合。证明

a) $A\cup B=B\cup A$　　b) $A\cap B=B\cap A$

8. 若 $A-B=\{1,5,7,8\}$，$B-A=\{2,10\}$，而 $A\cap B=\{3,6,9\}$，求集合 A 和 B。

9. 求证若 A 和 B 为集合，则 $\overline{A\cup B}=\overline{A}\cap\overline{B}$。

a) 用证明等式两边互为子集的办法。

b) 用成员表。

10. 令 A，B 为集合。求证

a) $(A\cap B)\subseteq A$　　b) $A\subseteq(A\cup B)$

c) $A-B\subseteq A$　　d) $A\cap(B-A)=\emptyset$

e) $A\cup(B-A)=A\cup B$

11. 设 A，B，C 为集合。证明 $\overline{A\cap B\cap C}=\overline{A}\cup\overline{B}\cup\overline{C}$。

12. 令 A，B，C 为集合。求证

a) $(A\cup B)\subseteq(A\cup B\cup C)$

b) $(A\cap B\cap C)\subseteq(A\cap B)$

c) $(A-B)-C\subseteq A-C$

d) $(A-C)\cap(C-B)=\emptyset$

e) $(B-A)\cup(C-A)=(B\cup C)-A$

13. 如果 A 和 B 为集合，求证 $A-B=A\cap\overline{B}$。

14. 如果 A 和 B 为集合，求证 $(A\cap B)\cup(A\cap\overline{B})=A$。

15. 令 A，B，C 为集合。求证

a) $A\cup(B\cup C)=(A\cup B)\cup C$

b) $A\cap(B\cap C)=(A\cap B)\cap C$

c) $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$

16. 令 A，B，C 为集合。求证 $(A-B)-C=(A-C)-(B-C)$。

17. 令 $A=\{0,2,4,6,8,10\}$, $B=\{0,1,2,3,4,5,6\}$, $C=\{4,5,6,7,8,9,10\}$。求

 a) $A\cap B\cap C$ b) $A\cup B\cup C$

 c) $(A\cup B)\cap C$ d) $(A\cap B)\cup C$

18. 画出集合 A，B，C 的下列组合的文氏图。

 a) $A\cap(B\cup C)$ b) $\overline{A}\cap\overline{B}\cap\overline{C}$

 c) $(A-B)\cup(A-C)\cup(B-C)$

19. 如果有关集合 A 与 B 的下列性质为真，你能就 A 和 B 说些什么？

 a) $A\cup B=A$ b) $A\cap B=A$

 c) $A-B=A$ d) $A\cap B=B\cap A$

 e) $A-B=B-A$

20. 如果集合 A，B，C 满足下述条件，能断定 $A=B$ 吗？

 a) $A\cup C=B\cup C$

 b) $A\cap C=B\cap C$

21. 令 A 和 B 为全集 U 的子集。求证 $A\subseteq B$ 当且仅当 $\overline{B}\subseteq\overline{A}$。

集合 A 和 B 的对称差，用 $A\oplus B$ 表示，是属于 A 或属于 B 但不同时属于 A 与 B 的元素组成的集合。

22. 求 $\{1,3,5\}$ 和 $\{1,2,3\}$ 的对称差。

23. 求某校主修计算机科学的学生集合与主修数学的学生集合的对称差。

24. 画出集合 A 与 B 的对称差的文氏图。

25. 证明 $A\oplus B=(A\cup B)-(A\cap B)$。

26. 证明 $A\oplus B=(A-B)\cup(B-A)$。

27. 求证若 A 是全集 U 的子集，则

 a) $A\oplus A=\emptyset$ b) $A\oplus\emptyset=A$

 c) $A\oplus U=\overline{A}$ d) $A\oplus\overline{A}=U$

28. 如果 A 和 B 为集合，求证

 a) $A\oplus B=B\oplus A$ b) $(A\oplus B)\oplus B=A$

29. 如果 $A\oplus B=A$，你能就集合 A 和 B 说些什么？

*30. 判断对称差是否满足结合律；即若 A，B，C 为集合，

$$A\oplus(B\oplus C)=(A\oplus B)\oplus C$$

是否成立？

*31. 假定 A，B，C 为集合，使得 $A\oplus C=B\oplus C$。是否必定有 $A=B$？

32. 若 A，B，C，D 为集合，$(A\oplus B)\oplus(C\oplus D)=(A\oplus C)\oplus(B\oplus D)$是否成立？

33. 若 A，B，C，D 为集合，$(A\oplus B)\oplus(C\oplus D)=(A\oplus D)\oplus(B\oplus C)$是否成立？

*34 求证：若 A，B，C 为集合，则

$$|A\cup B\cup C|=|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|$$

（这是第 5 章将要学习的包含排斥原理的一个特例。）

35. 令 $A_i=\{1,2,3,\cdots,i\}$, $i=1,2,3,\cdots$。求

a) $\bigcup_{i=1}^{n} A_i$ b) $\bigcap_{i=1}^{n} A_i$

36. 令 $A_i = \{i, i+1, i+2, \cdots\}$。求

a) $\bigcup_{i=1}^{n} A_i$ b) $\bigcap_{i=1}^{n} A_i$

37. 令 A_i 为所有长度不超过 i 的非空位串（即长度至少为 1）的集合。求

a) $\bigcup_{i=1}^{n} A_i$ b) $\bigcap_{i=1}^{n} A_i$

38. 假定全集 $U = \{1,2,3,4,5,6,7,8,9,10\}$。用位串表示下列各集合，假定若 i 属于该集合则其位串第 i 位为 1，否则第 i 位为 0。

a) $\{3,4,5\}$ b) $\{1,3,6,10\}$

c) $\{2,3,4,7,8,9\}$

39. 使用上题中的同一个全集，求下列位串各自代表的集合。

a) 11 1100 1111 b) 01 0111 1000

c) 10 0000 0001

40. 下列位串各代表有限全集的什么子集？

a) 所有字位全为 0 的串。

b) 所有字位全为 1 的串。

41. 对应于两个集合之差的位串是什么？

42. 对应于两个集合的对称差的位串是什么？

43. 令 $A = \{a,b,c,d,e\}$, $B = \{b,c,d,g,p,t,v\}$, $C = \{c,e,i,o,u,x,y,z\}$, $D = \{d,e,h,i,n,o,t,u,x,y\}$。说明怎样用位串的按位运算求下列集合：

a) $A \cup B$ b) $A \cap B$

c) $(A \cup B) \cap (B \cup C)$ d) $A \cup B \cup C \cup D$

44. 怎样用位串求出同一全集 U 的 n 个子集的并集和交集？

45. 集合 A 的后继是 $A \cup \{A\}$。求下列集合的后继。

a) $\{1,2,3\}$ b) $\emptyset$

c) $\{\emptyset\}$ d) $\{\emptyset, \{\emptyset\}\}$

46. 含 n 个元素的集合的后继有几个元素？

有时一个元素在一个无序集中出现的次数也有意义。当同一个元素作为成员可以出现不止一次时，这个无序元素集就是**多重集**。符号 $\{m_1 \cdot a_1, m_2 \cdot a_2, \cdots, m_r \cdot a_r\}$ 表示的是 a_1 出现 m_1 次，a_2 出现 m_2 次等的多重集。整数 m_i 称为 a_i 的重数，$i = 1, 2, \cdots, r$。

令 P 和 Q 为多重集，多重集 P 和 Q 的并集是个多重集，其中每个元素的重数是该元素在 P 和 Q 中的重数的最大值。P 和 Q 的交集是个多重集，其中每个元素的重数是该元素在 P 和 Q 中重数的最小值。P 和 Q 的差集是个多重集，其中每个元素的重数是该元素在 P 中的重数减去它在 Q 中的重数，当然这不能是个负数，若是负数就以 0 作重数。P 和 Q 的和集是个多重集，其中每个元素的重数是该元素在 P 和 Q 中的重数的和，P 和 Q 的并集，交集和差集分别用 $P \cup Q$，$P \cap Q$ 和 $P - Q$ 表示（不要混淆这些运算和集合上的类似运算）。P 和 Q 的和集用 $P + Q$ 表示。

47. 令 A 和 B 分别为多重集 $\{3 \cdot a, 2 \cdot b, 1 \cdot c\}$ 和 $\{2 \cdot a, 3 \cdot b, 4 \cdot d\}$。求

a) $A \cup B$　　b) $A \cap B$　　c) $A - B$
d) $B - A$　　e) $A + B$

48. 假定 A 是个多重集，其元素是某大学一个系需要的设备的类型，而元素的重数则是该类设备的件数；B 是同一所大学另一个系类似的多重集。例如 A 可以是多重集 {107·PC, 44·routers, 6·servers}，而 B 可以是 {14·PC, 6·routers, 2·mainframes}，其中 routers 为路由器，servers 为服务器，而 mainframes 为大型计算机。
    a) 假定两个系使用同一个设备，A 和 B 的什么组合代表该大学应该买的设备？
    b) 假定两个系使用同一个设备，A 和 B 的什么组合代表两个系都使用的设备？
    c) 假定两个系使用同一个设备，A 和 B 的什么组合代表第二个系使用，但第一个系不使用的设备？
    d) 假定两个系不共享设备，A 和 B 的什么组合代表该大学应该购买的设备？

人工智能中使用模糊集合。全集 U 中每个元素在模糊集合 S 中都有个成员度，即 0 和 1 之间（包括 0 和 1）的实数。模糊集合 S 的表示法是列出其元素及成员度（成员度为 0 的元素不列）。例如，用 {0.6 Alice, 0.9 Brian, 0.4 Fred, 0.1 Oscar, 0.5 Rita} 表示名人集合 F，说明 F 中 Alice 的成员度为 0.6，Brian 为 0.9，Fred 为 0.4，Oscar 为 0.1，而 Rita 为 0.5（因此这些人里而 Brian 最出名而 Oscar 最不出名）。再假定 R 是富人集合，R = {0.4 Alice, 0.8 Brian, 0.2 Fred, 0.9 Oscar, 0.7 Rita}。

49. 模糊集合 S 的补是集合 $\overline{S}$，元素在 $\overline{S}$ 中的成员度等于 1 减去该元素在 S 中的成员度。求 $\overline{F}$（不出名者的模糊集合）和 $\overline{S}$（不富裕者的模糊集合）。
50. 模糊集合 S 和 T 的并集是模糊集合 $S \cup T$，其中每个元素的成员度是该元素在 S 和 T 中成员度的最大值。求名人或富人的模糊集合 $F \cup R$。
51. 模糊集合 S 和 T 的交集是模糊集合 $S \cap T$，其中每个元素的成员度是该元素在 S 和 T 中的成员度的最小值。求既出名又富裕者的模糊集合 $F \cap R$。

1.6 函数

1.6.1 引言

在许多情况下我们都会为一个集合的每个元素指派另一个集合（可以就是第一个集合）的一个特定元素。例如，假定对离散数学课的每个学生从 {A, B, C, D, F} 中选一个字母作为他的得分。再假定 Adams 的得分是 A，Chou 的得分是 C，Goodfriend 的得分是 B，Rodriguez 的得分是 A，而 Stevens 的得分是 F。这一打分如图 1-8 所示。

这种打分就是一个函数。离散数学中函数的概念分外重要。在定义像序列和字符串这样的离散结构时，用到函数。函数还用于表示解决一定规模的问题需要的计算机机时。借助自身定义的函数称为递归函数。计算机科学处处使用的这种函数将在第 3 章再讨论。这一节只回顾与离散数学需要有关的函数的基本概念。

定义 1　令 A 和 B 为集合。从 A 到 B 的函数 f 是对元素的一种指派，对 A 的每个元素恰好指派的 B 的一个元素。如果 f 指派给 A 中元素 a 的唯一的 B 元素是 b，就写成 $f(a) = b$。如果 f 是从 A 到 B 的函数，就写成 f: A→B。

有许多描述函数的方式。有时明确说明指派关系，但常用的方式是给出指派公式来定义函数，如$f(x)=x+1$。有时也用计算机程序来描述函数。

定义 2　如果 f 是从 A 到 B 的函数，就说 A 是 f 的定义域，而 B 是 f 的伴域。如果 $f(a)=b$，就说 b 是 a 的像而 a 是 b 的原像。A 中元素的所有像元素的集合称为 f 的值域。若 f 是从 A 到 B 的函数，有时也说成 f 把 A 映射到 B。

图 1-9 表示一个从 A 到 B 的函数。

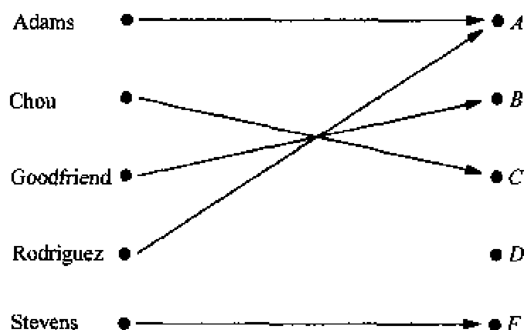


图 1-8　离散数学课成绩

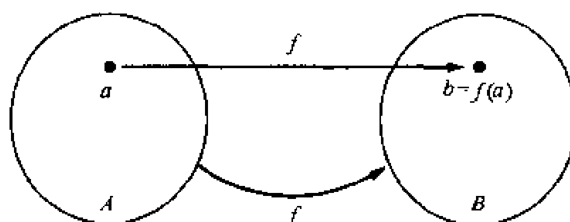


图 1-9　函数 f 把 A 映射到 B

考虑本节开头的例子。令 G 为给离散数学课每个学生打分的函数。例如，G(Adams) = A。G 的定义域是集合 {Adams，Chou，Goodfriend，Rodriguez，Stevens}，而伴域则是集合 {A，B，C，D，F}。G 的值域是集合 {A，B，C，F}，因为除 D 以外的分值都有学生获得。再考虑下面几个例子。

例 1　令 f 为这样的函数，它把长度为 2 或大于 2 的位串的最后两个字位指派给该位串。于是 f 的定义域是长度不小于 2 的所有位串，而伴域和值域均为集合 {00，01，10，11}。■

例 2　令 f 为从 **Z** 到 **Z** 的函数，它指派给每个整数的是该整数的平方。于是 $f(x)=x^2$，而 f 的定义域是所有整数的集合，f 的伴域可以从所有整数的集合中选择，f 的值域是所有非负整数中那些完全平方的集合，即$\{0,1,4,9,\cdots\}$。■

例 3　（适合于熟悉 Pascal 的学生）函数的定义域和伴域往往以程序语言描述。例如 Pascal 语句

```
function floor(x:real):integer
```

说的是，floor 函数的定义域是实数集合，而它的伴域是整数集合。■

具有相同定义域的两个实数值函数可以相加和相乘。

定义 3　令 f_1 和 f_2 是从 A 到 **R** 的函数，那么 f_1+f_2 和 f_1f_2 也是从 A 到 **R** 的函数，其定义为

$$(f_1+f_2)(x)=f_1(x)+f_2(x)$$
$$(f_1f_2)(x)=f_1(x)f_2(x)$$

注意 f_1+f_2 和 f_1f_2 的定义是利用 f_1 和 f_2 在 x 的值计算它们在 x 的值。

例4 令 f_1 和 f_2 是从 **R** 到 **R** 的函数，且 $f_1(x)=x^2$ 而 $f_2(x)=x-x^2$。函数 f_1+f_2 和 f_1f_2 是什么？

解 从函数的和与积的定义知

$$(f_1+f_2)(x)=f_1(x)+f_2(x)=x^2+(x-x^2)=x$$

而

$$(f_1f_2)(x)=x^2(x-x^2)=x^3-x^4$$ ■

若 f 是从集合 A 到集合 B 的函数，则可以定义 A 的子集的像。

定义4 令 f 为从集合 A 到集合 B 的函数，S 为 A 的一个子集。S 的像是由 S 中元素的像组成的 B 的子集。我们用 $f(S)$ 表示 S 的像，于是

$$f(S)=\{f(s)\mid s\in S\}$$

例5 令 $A=\{a,b,c,d,e\}$ 而 $B=\{1,2,3,4\}$，且 $f(a)=2, f(b)=1, f(c)=4, f(d)=1$ 及 $f(e)=1$。子集 $S=\{b,c,d\}$ 的像是集合 $f(S)=\{1,4\}$。 ■

1.6.2 一对一函数和映上函数

有些函数在它们的定义域的不同成员上有不同的像。这种函数称为一对一的。

定义5 函数 f 称为一对一的或单射的，当且仅当对于 f 的定义域中的所有 x 和 y，$f(x)=f(y)$ 蕴含着 $x=y$。一对一的函数称为单射。

注意 函数 f 是一对一的，当且仅当只要 $x\neq y$ 就有 $f(x)\neq f(y)$。这种表达 f 为一对一函数的方式是对定义中的蕴含倒置而来。

例6 判断从 $\{a,b,c,d\}$ 到 $\{1,2,3,4,5\}$ 的函数 f 是否为一对一的，f 的定义是 $f(a)=4, f(b)=5, f(c)=1$ 而 $f(d)=3$。

解 f 是一对一的，因为 f 在它定义域的四个元素上取不同的值。图1-10说明了这一点。 ■

例7 判断从整数集合到整数集合的函数 $f(x)=x^2$ 是否为一对一的。

解 函数 $f(x)=x^2$ 不是一对一的，因为，例如 $f(1)=f(-1)=1$，但 $1\neq -1$。 ■

例8 判断函数 $f(x)=x+1$ 是否为一对一的。

解 函数 $f(x)=x+1$ 是一对一的。要证明这一点，只须注意在 $x\neq y$ 时 $x+1\neq y+1$。 ■

现在我们给出保证函数为一对一的某些条件。

定义6 定义域和伴域都是实数集合子集的函数 f 称为**严格递增的**，如果对 f 的定义域中的 x 和 y，只要 $x<y$ 就有 $f(x)<f(y)$。类似地，f 称为**严格递减的**，如果对 f 定义域中的 x 和 y，只要 $x<y$ 就有 $f(x)>f(y)$。

从上述定义可知，只要函数是严格递增的或严格递减的，它必定是一对一的。

有些函数的值域和伴域相等，也就是说伴域中的每个成员都是定义域中某个元素的像。具有这一性质的函数称为映上函数。

定义 7　从 A 到 B 的函数 f 称为映上的或满射的，当且仅当对每个 $b \in B$，有元素 $a \in A$ 使得 $f(a)=b$。如果函数 f 是映上的，就说它是满射函数。

我们现在举几个映上函数和非映上函数的例子。

例 9　令 f 为从 $\{a, b, c, d\}$ 到 $\{1,2,3\}$ 的函数，其定义为 $f(a)=3, f(b)=2, f(c)=1$ 及 $f(d)=3$。f 是映上函数吗？

解　由于伴域中所有 3 个元素均为定义域中元素的像，f 是映上的。图 1-11 说明了这一点。■

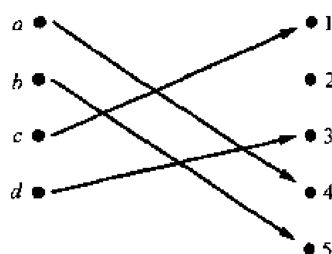


图 1-10　一个一对一函数

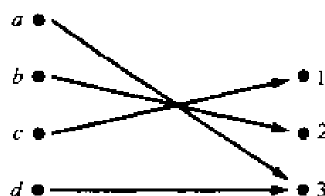


图 1-11　一个映上函数

例 10　从整数集到整数集的函数 $f(x)=x^2$ 是映上的吗？

解　f 不是映上的，因此，比如说没有 x 使 $f(x)=-1$。■

例 11　从整数集到整数集的函数 $f(x)=x+1$ 是映上的吗？

解　这个函数是映上的，因为对每个整数 y，都有一个整数 x 使 $f(x)=y$。为看出这一点，只要注意 $f(x)=y$ 的充分必要条件是 $x+1=y$，而这只要令 $x=y-1$ 就成立。■

定义 8　若函数 f 既是一对一的，又是映上的，就说它是一一对应或双射的。

下面几个例子阐述双射的概念。

例 12　令 f 为从 $\{a, b, c, d\}$ 到 $\{1,2,3,4\}$ 的函数，其定义为 $f(a)=4, f(b)=2, f(c)=1$ 及 $f(d)=3$。f 是双射吗？

解　函数 f 是一对一的和映上的。它是一对一的，是因为函数值都不同；它是映上的，是因为伴域中的所有 4 个元素，均为定义域的元素的像。于是 f 是双射。■

图 1-12 给出了 4 个函数：其中第 1 个是一对一的，但不是映上的；第 2 个是映上的，但不是一对一的；第 3 个，既是一对一的，又是映上的；第 4 个既不是一对一的，也不是映上的。图 1-12 中的第 5 个对应关系不是函数，因为它把一个元素传递给两个不同的元素。

假定 f 是从集合 A 到它自己的函数。如果 A 是有限的，那么 f 是一对一的当且仅当它是映上的。（从本节末练习 58 的结果即可得出这一结论。）当 A 为无限集时，这一结论不一定成立（参看 1.7 节）。

例 13　令 A 为集合。A 上的恒等函数是函数 $\iota_A: A \to A$，其中

$$\iota_A(x)=x$$

对所有 $x \in A$。换言之，恒等函数 ι_A 是这样的函数，它赋给每个元素的是这个元素自身。函数 ι_A 是一对一的和映上的，所以是双射。

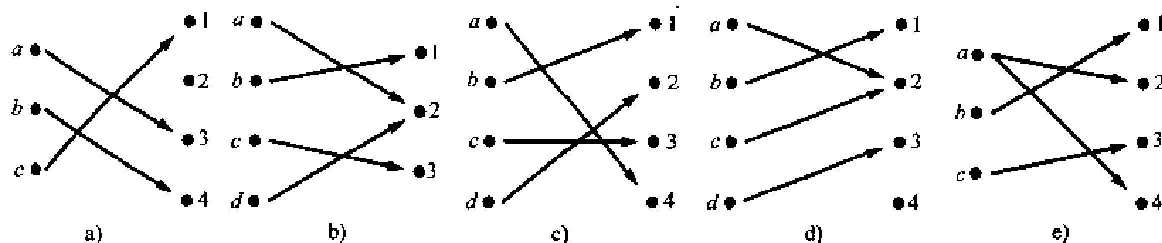


图 1-12 不同类型的对应关系的例子

a) 一对一，非映上；b) 映上，非一对一；c) 一对一，映上；d) 既非一对一，也非映上；e) 不是函数

1.6.3 反函数和函数组合

现在考虑从集合 A 到集合 B 的一一对应 f。由于 f 是映上的函数，B 的每个元素都是 A 中某元素的像。又由于 f 还是一对一的函数，B 的每个元素都是 A 中唯一的一个元素的像。于是我们可以定义一个从 B 到 A 的新函数，把 f 给出的对应关系颠倒过来。

定义 9 令 f 为从集合 A 到集合 B 的一一对应，f 的反函数是这样的函数，它指派给 B 中元素 b 的是 A 中使得 $f(a)=b$ 唯一元素 a。f 的反函数用 f^{-1} 表示，于是在 $f(a)=b$ 时 $f^{-1}(b)=a$。

图 1-13 给出了反函数概念的说明。

如果函数 f 不是一一对应，就无法定义反函数。若 f 不是一一对应，那么它或者不是一对一的，或者不是映上的，如果 f 不是一对一的，则伴域中的某元素 b 是定义域中多个元素的像。如果 f 不是映上的，那么伴域中某个元素 b 不是定义域中任何元素的像，即定义域中不存在元素 a 使 $f(a)=b$。总之如果 f 不是一一对应，就不能为伴域中每个元素 b，都指派定义域中唯一的元素 a，使 $f(a)=b$（因为对某个 b 或者有多个这样的 a，或者没有这样的 a）。

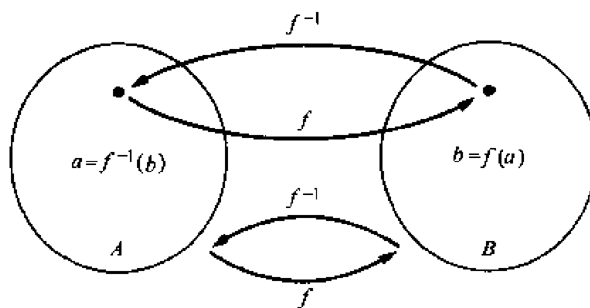


图 1-13 函数 f^{-1} 是函数 f 的反函数

一一对应关系称为*可逆的*，因为作为函数可以定义它的反函数。如果函数不是一一对应关系，就说它是*不可逆的*，因为这样的函数没有反函数。

例 14 令 f 为从 $\{a,b,c\}$ 到 $\{1,2,3\}$ 的函数，使 $f(a)=2, f(b)=3$ 及 $f(c)=1$。f 可逆吗？如果可逆，其反函数是什么？

解 f 是可逆的，因为它是一个一对一的对应关系。其反函数 f^{-1} 颠倒 f 给出的对应关系，所以 $f^{-1}(1)=c, f^{-1}(2)=a$ 而 $f^{-1}(3)=b$。■

例 15 令 f 为从整数集到整数集的函数，使得 $f(x)=x+1$。f 可逆吗？如果可逆，其

反函数是什么？

解 f 可逆，因为前面已证明它是一一对应关系。要颠倒对应关系，设 y 是 x 的像，即 $y=x+1$。从而 $x=y-1$。即 $y-1$ 是 **Z** 的唯一元素，在 f 之下与 y 对应，于是 $f^{-1}(y)=y-1$。 ■

例 16 令 f 是从 **Z** 到 **Z** 的函数，使 $f(x)=x^2$。f 可逆吗？

解 由于 $f(-1)=f(1)=1$，f 不是一对一的，要想定义反函数，就得为 1 指派两个元素。因此 f 是不可逆的。 ■

定义 10 令 g 为从集合 A 到集合 B 的函数，f 是从集合 B 到集合 C 的函数，函数 f 和 g 的组合用 $f\circ g$ 表示，定义为

$$(f\circ g)(a)=f(g(a))$$

换句话说，函数 $f\circ g$ 指派给 A 的元素 a 的就是 f 指派给 $g(a)$ 的。注意，如果 g 的值域不是 f 的定义域的子集，就无法定义 $f\circ g$，图 1-14 解释了函数组合的概念。

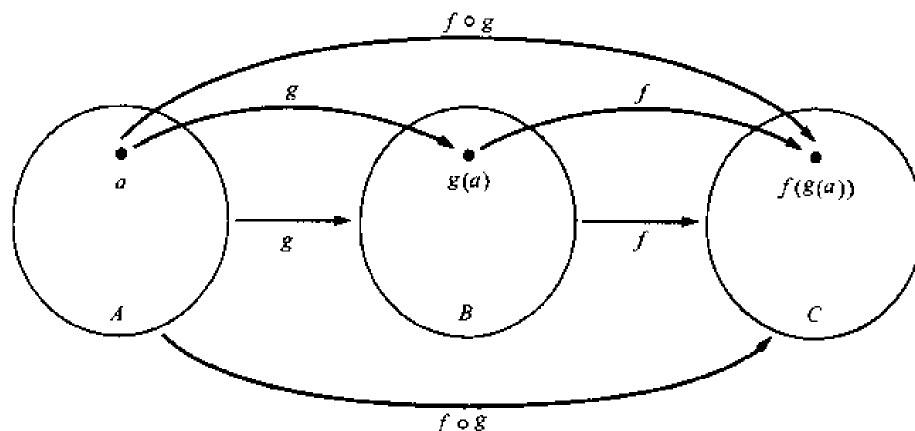


图 1-14 函数 f 和 g 的组合

例 17 令 g 为从 $\{a,b,c\}$ 到它自己的函数，使 $g(a)=b, g(b)=c$，而 $g(c)=a$。令 f 为从 $\{a,b,c\}$ 到 $\{1,2,3\}$ 的函数，使 $f(a)=3, f(b)=2$，而 $f(c)=1$。f 和 g 的组合是什么？g 和 f 的组合是什么？

解 $f\circ g$ 的定义是 $(f\circ g)(a)=f(g(a))=f(b)=2$，$(f\circ g)(b)=f(g(b))=f(c)=1$，而 $(f\circ g)(c)=f(g(c))=f(a)=3$。 ■

注意 $g\circ f$ 是没有定义的，因为 f 的值域不是 g 的定义域的一部分。

例 18 令 f 和 g 为从整数集到整数集的函数，其定义为 $f(x)=2x+3$ 和 $g(x)=3x+2$。f 和 g 的组合是什么？g 和 f 的组合是什么？

解 $f\circ g$ 和 $g\circ f$ 均有定义，即

$$(f\circ g)(x)=f(g(x))=f(3x+2)=2(3x+2)+3=6x+7$$

及

$$(g\circ f)(x)=g(f(x))=g(2x+3)=3(2x+3)+2=6x+11$$ ■

注意　尽管例 18 中对函数 f 和 g 而言 $f\circ g$ 和 $g\circ f$ 均有定义，$f\circ g$ 和 $g\circ f$ 并不相等。换言之，对函数组合而言交换律不成立。

在构造函数和它的反函数的组合时，不论以什么次序组合，得到的都是恒等函数。要看清这一点，假定 f 是从集合 A 到集合 B 的一一对应关系，那么存在反函数 f^{-1}，而且 f^{-1}是从 B 到 A 的一一对应关系。反函数把原函数的对应关系颠倒过来，所以当 $f(a)=b$ 时 $f^{-1}(b)=a$，当 $f^{-1}(b)=a$ 时，$f(a)=b$。因此，

$$(f^{-1}\circ f)(a)=f^{-1}(f(a))=f^{-1}(b)=a$$

及

$$(f\circ f^{-1})(b)=f(f^{-1}(b))=f(a)=b$$

从而 $f^{-1}\circ f=\iota_A$，而 $f\circ f^{-1}=\iota_B$，其中 ι_A 和 ι_B 分别是集合 A 和 B 上的恒等函数。这就是说 $(f^{-1})^{-1}=f$。

1.6.4　函数的图像

从 A 到 B 的每个函数都对应着 $A\times B$ 中的一个有序偶集合。这个序偶集合称为该函数的图像，并且往往用图表示以帮助理解函数的行为。

定义 11　令 f 为从集合 A 到集合 B 的函数，函数 f 的图像是序偶集合 $\{(a,b)\mid a\in A$ 且 $f(a)=b\}$。

根据定义，从 A 到 B 的函数的图像是 $A\times B$ 的子集，这个子集包含的序偶中第二项等于 B 中由 f 指派给第一项的那个元素。

例 19　图示从整数集到整数集的函数 $f(n)=2n+1$ 的图像。

解　f 的图像是形为 $(n,2n+1)$ 的序偶的集合，其中 n 为整数。图 1-15 是其图像显示。■

例 20　图示整数集到整数集的函数 $f(x)=x^2$。

解　f 的图像是形为 $(x,f(x))=(x,x^2)$ 的序偶的集合，其中 x 为整数。图1-16 是其图像显示。■

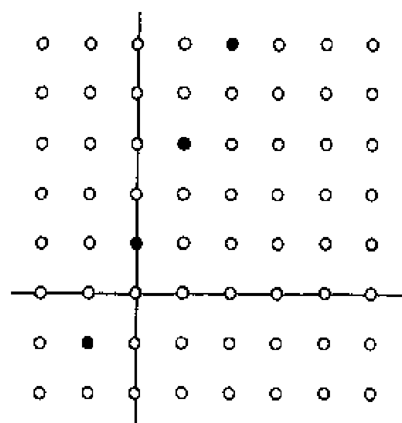


图 1-15　从 Z 到 Z 的函数 $f(n)=2n+1$ 的图像

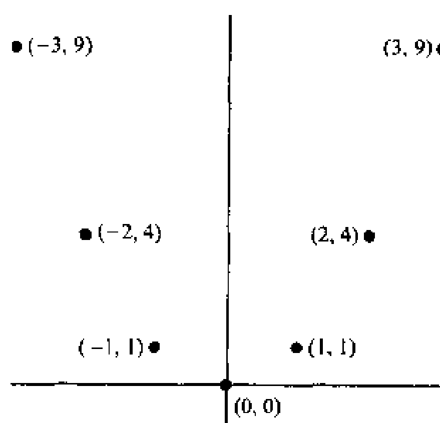


图 1-16　从 Z 到 Z 的 $f(x)=x^2$ 的图像

1.6.5　几个重要的函数

下面介绍离散数学中两个重要的函数，即底函数和顶函数。令 x 为实数。底函数把 x 下舍入到小于或等于 x 又最接近 x 的整数，而顶函数则把 x 上舍入到大于或等于 x 又最接近 x 的整数。在统计对象个数时常使用这两个函数。在分析解一定规模的问题的计算机过程使用的步数时，这两个函数起着重要的作用。

定义 12　底函数指派给实数 x 的是小于或等于 x 的最大整数。底函数在 x 的值用 $\lfloor x\rfloor$ 表示。顶函数指派给实数 x 的是大于或等于 x 的最小整数。顶函数在 x 的值用 $\lceil x\rceil$ 表示。

注意　*底函数也常称为最大整数函数，这时往往用 $[x]$ 表示。*

例 21　下面是底函数和顶函数的若干值：

$$\lfloor 1/2\rfloor=0,\quad \lceil 1/2\rceil=1,\quad \lfloor -1/2\rfloor=-1,\quad \lceil -1/2\rceil=0,$$
$$\lfloor 3.1\rfloor=3,\quad \lceil 3.1\rceil=4,\quad \lfloor 7\rfloor=7,\quad \lceil 7\rceil=7$$ ■

图 1-17 显示的是底函数和顶函数的图像。

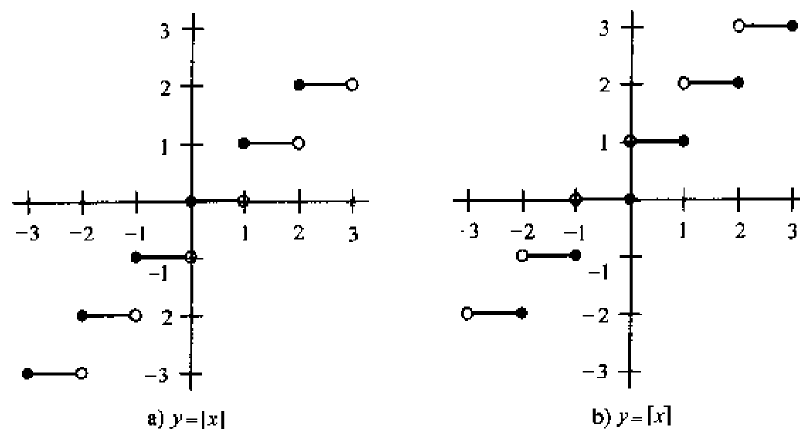


图 1-17　a) 底函数图像　b) 顶函数图像

底函数和顶函数有广泛的应用，包括用于数据存储和数据传输。考虑下面的例子，这是研究数据库和数据通信问题时要完成的典型的基本计算。

例 22　存在计算机磁盘上的数据或数据网络上传输的数据通常表示为字节串。每个字节由 8 个字位组成，要表示 100 字位的数据需要多少字节？

解　要决定需要的字节数，就要找出最小的整数，它至少要与 100 除以 8 的商一样大，8 是每个字节的字位数。于是，需要的字节数是 $\lceil 100/8\rceil=\lceil 12.8\rceil=13$。 ■

例 23　异步传输模式（ATM）（用于骨干网络上的通信协议）下，数据按 53 个字节分组，每组称为一个信元。以速率每秒 500 千字位传输数据的连接上一分钟能传输多少个 ATM 信元？

解　一分钟内这一连接能传输 $500\ 000\times 60=30\ 000\ 000$ 字位。每个 ATM 信元长度是 53 字节，也就是 $53\times 8=424$ 字位。要计算一分钟能传输多少信元，需计算不超过

30 000 000除以 424 的最大整数。于是在每秒 500 千字位的连接上一分钟能传输的 ATM 信元数是$\lfloor 30\ 000\ 000/424 \rfloor = 70\ 754$。 ■

以 x 代表实数的表 1-19 给出了底函数和项函数的若干简单但重要的性质。由于这两个函数在离散数学中出现得十分频繁，看一看表中的恒等式是有益的。表中的每条性质都可以用底函数和顶函数的定义来建立。性质（1a），（1b），（1c）和（1d）可以直接由定义得出。例如，（1a）说的是$\lfloor x \rfloor = n$ 当且仅当 n 小于或等于 x 而 $n+1$ 大于 x。这恰恰就是 n 为不超过 x 的最大整数的含义，也就是$\lfloor x \rfloor = n$ 的定义。性质（1b），（1c）和（1d）可以类似地建立起来。

我们来证明（4a）是成立的。为此，假定$\lfloor x \rfloor = n$，其中 n 为整数。由（1a）知 $n \leqslant x < n+1$。在这一不等式中加上 m 得 $n+m \leqslant x+m < n+m+1$。再次利用（1a），可知$\lfloor x+m \rfloor = n+m = \lfloor x \rfloor + m$，这正是我们要证明的。其他性质的证明推迟到练习中去做。

有几类函数是本课本一直要用到的，其中包括多项式，对数和指数函数。附录 1 给出了课本中需要的这些函数之性质的简要回顾。本书中用记号 $\log x$ 表示 x 以 2 为基的对数，因为 2 是我们将经常使用的对数基数。我们用 $\log_b x$ 表示以 b 为基的对数，其中 b 是大于 1 的任意实数。

表 1-19 顶函数和底函数的有用性质

(1a) $\lfloor x \rfloor = n$ 当且仅当 $n \leqslant x < n+1$，其中 n 为整数
(1b) $\lceil x \rceil = n$ 当且仅当 $n-1 < x \leqslant n$，其中 n 为整数
(1c) $\lfloor x \rfloor = n$ 当且仅当 $x-1 < n \leqslant x$，其中 n 为整数
(1d) $\lceil x \rceil = n$ 当且仅当 $x \leqslant n < x+1$，其中 n 为整数
(2) $x-1 < \lfloor x \rfloor \leqslant x \leqslant \lceil x \rceil < x+1$
(3a) $\lfloor -x \rfloor = -\lceil x \rceil$
(3b) $\lceil -x \rceil = -\lfloor x \rfloor$
(4a) $\lfloor x+m \rfloor = \lfloor x \rfloor + m$，$m$ 为整数
(4b) $\lceil x+m \rceil = \lceil x \rceil + m$，$m$ 为整数

练习

1. 为什么下列几间中的 f 不是从 **R** 到 **R** 的函数？
   a) $f(x) = 1/x$　　b) $f(x) = \sqrt{x}$
   c) $f(x) = \pm\sqrt{(x^2+1)}$
2. 判断下面定义的几个 f 是不是从 **Z** 到 **R** 的函数。
   a) $f(n) = \pm n$　　b) $f(n) = \sqrt{(n^2+1)}$
   c) $f(n) = 1/(n^2-4)$
3. 判断 f 是否为从所有位串的集合到整数集合的函数：
   a) $f(S)$是 S 中某个 0 字位的位置。
   b) $f(S)$是 S 中 1 字位的个数。
   c) $f(S)$是最小整数 i 使 S 中的第 i 字位为 1，当 S 是不含字位的空串时$f(S)=0$。

4. 求下列函数的定义域和值域。

a) 函数在每个非负整数的值为该整数的最后一个数字。

b) 函数在每个正整数的值是比它小的最大整数。

c) 函数在每个位串的值是串中 1 字位的个数。

d) 函数在每个位串的值是串的字位数。

5. 求下列函数的定义域和值域。

a) 函数在每个位串的值是串中 1 字位个数与 0 字位个数之差。

b) 函数在每个位串的值是串中 0 字位个数的两倍。

c) 函数在每个位串的值是把串分成字节（8 个字位为一字节）时不够一个字节的字位数。

d) 函数在每个正整数的值是不超过此整数的最大完全平方。

6. 求下列各值：

a) $\lfloor 1.1 \rfloor$　　b) $\lceil 1.1 \rceil$

c) $\lfloor -0.1 \rfloor$　　d) $\lceil -0.1 \rceil$

e) $\lceil 2.99 \rceil$　　f) $\lceil -2.99 \rceil$

g) $\lfloor 1/2 + \lceil 1/2 \rceil \rfloor$　　h) $\lceil \lfloor 1/2 \rfloor + \lceil 1/2 \rceil + 1/2 \rceil$

7. 求下列各值：

a) $\lceil 3/4 \rceil$　　b) $\lfloor 7/8 \rfloor$

c) $\lceil -3/4 \rceil$　　d) $\lfloor -7/8 \rfloor$

e) $\lceil 3 \rceil$　　f) $\lfloor -1 \rfloor$

g) $\lfloor 1/2 + \lceil 3/2 \rceil \rfloor$　　h) $\lfloor 1/2 \cdot \lfloor 5/2 \rfloor \rfloor$

8. 判断下列从 $\{a, b, c, d\}$ 到它自身的函数是否一对一。

a) $f(a) = b, f(b) = a, f(c) = c, f(d) = d$

b) $f(a) = b, f(b) = b, f(c) = d, f(d) = c$

c) $f(a) = d, f(b) = b, f(c) = c, f(d) = d$

9. 练习 8 中哪些函数是映上的？

10. 判断下列 **Z** 到 **Z** 的函数是否一对一。

a) $f(n) = n - 1$　　b) $f(n) = n^2 + 1$

c) $f(n) = n^3$　　d) $f(n) = \lceil n/2 \rceil$

11. 上一题中那些函数是映上的？

12. 给出从 **N** 到 **N** 的函数的例子，使得：

a) 一对一但非映上。

b) 映上但不一对一。

c) 既映上又一对一（但不同于恒等函数）。

d) 既非映上又非一对一。

13. 给出从整数集合到正整数集合的函数的明确公式，使得：

a) 一对一但非映上。

b) 映上但非一对一。

c) 既映上又一对一。

d) 既不映上又不一对一。

14. 判断下列各函数是否从 **R** 到 **R** 的双射。
   a) $f(x)=-3x+4$　　b) $f(x)=-3x^2+7$
   c) $f(x)=(x+1)/(x+2)$　　d) $f(x)=x^5+1$
15. 判断下列各函数是否从 **R** 到 **R** 的双射。
   a) $f(x)=2x+1$　　b) $f(x)=x^2+1$
   c) $f(x)=x^3$　　d) $f(x)=(x^2+1)/(x^2+2)$
16. 令 $S=\{-1,0,2,4,7\}$。求 $f(S)$，如果
   a) $f(x)=1$　　b) $f(x)=2x+1$
   c) $f(x)=\lceil x/5\rceil$　　d) $f(x)=\lfloor(x^2+1)/3\rfloor$
17. 令 $f(x)=\lfloor x^2/3\rfloor$。求 $f(S)$，如果
   a) $S=\{-2,-1,0,1,2,3\}$　　b) $S=\{0,1,2,3,4,5\}$
   c) $S=\{1,5,7,11\}$　　d) $S=\{2,6,10,14\}$
18. 令 $f(x)=2x$。什么是
   a) $f(\mathbf{Z})$?　　b) $f(\mathbf{N})$?　　c) $f(\mathbf{R})$?
19. 假定 g 是从 A 到 B 的函数，f 是从 B 到 C 的函数。
   a) 证明如果 f 和 g 均为一对一函数，那么 $f\circ g$ 也是。
   b) 证明如果 f 和 g 均为到映上函数，那么 $f\circ g$ 也是。

*20. 如果 f 和 $f\circ g$ 都是一对一的，能否得出结论 g 也是一对一的？说明理由。

*21. 如果 f 和 $f\circ g$ 都是映上的，能否得出结论 g 也是映上的？说明理由。

22. 已知 $f(x)=x^2+1$ 和 $g(x)=x+2$ 都是从 **R** 到 **R** 的函数，求 $f\circ g$ 和 $g\circ f$。
23. 对上题中的函数 f 和 g 求 $f+g$ 和 fg。
24. 令 $f(x)=ax+b$，$g(x)=cx+d$，其中 a,b,c,d 为常数。哪些常数 a,b,c,d 能使 $f\circ g=g\circ f$?
25. 求证从 **R** 到 **R** 的函数 $f(x)=ax+b$ 是可逆的，其中 a 和 b 为常数且 $a\neq 0$。找出 f 的反函数。
26. 令 f 为从集合 A 到集合 B 的函数，令 S 和 T 为 A 的子集，求证
   a) $f(S\cup T)=f(S)\cup f(T)$
   b) $f(S\cap T)\subseteq f(S)\cap f(T)$
27. 给出一个例子说明上题 b) 中可能是真子集。

令 f 为从集合 A 到集合 B 的函数，S 为 B 的子集。S 的逆像定义为 A 的子集，该子集包含 S 的所有元素的原像。我们用 $f^{-1}(S)$ 表示 S 的逆像，于是 $f^{-1}(S)=\{a\in A\mid f(a)\in S\}$。

28. 令 f 为从 **R** 到 **R** 的函数 $f(x)=x^2$。求
   a) $f^{-1}(\{1\})$　　b) $f^{-1}(\{x\mid 0<x<1\})$
   c) $f^{-1}(\{x\mid x>4\})$
29. 令 $g(x)=\lfloor x\rfloor$。求
   a) $g^{-1}(\{0\})$　　b) $g^{-1}(\{-1,0,1\})$
   c) $g^{-1}(\{x\mid 0<x<1\})$

30. 令 f 为从 A 到 B 的函数，令 S 和 T 为 B 的子集，求证
    a) $f^{-1}(S\cup T)=f^{-1}(s)\cup f^{-1}(T)$
    b) $f^{-1}(S\cap T)=f^{-1}(S)\cap f^{-1}(T)$
31. 令 f 为从 A 到 B 的函数，S 为 B 的子集。求证 $f^{-1}(\overline{S})=\overline{f^{-1}(S)}$。
32. 证明 $\lfloor x+1/2\rfloor$ 是最接近 x 的整数，除非 x 恰为两个（相邻）整数的中间数，此时它为这两个整数中较大的一个。
33. 证明 $\lceil x-1/2\rceil$ 是最接近 x 的整数，除非 x 恰为两个（相邻）整数的中间数，此时，它为这两个整数中较大的一个。
34. 证明对实数 x，当 x 不是整数时 $\lceil x\rceil-\lfloor x\rfloor=1$，而在 x 为整数时 $\lceil x\rceil-\lfloor x\rfloor=0$。
35. 证明对实数 x，$x-1<\lfloor x\rfloor\leqslant x\leqslant x\leqslant\lceil x\rceil<x+1$。
36. 若 x 为实数，m 为整数，求证 $\lceil x+m\rceil=\lceil x\rceil+m$。
37. 若 x 为实数，n 为整数，求证
    a) $x<n$ 当且仅当 $\lfloor x\rfloor<n$。
    b) $n<x$ 当且仅当 $n<\lceil x\rceil$。
38. 若 x 为实数，n 为整数，求证
    a) $x\leqslant n$ 当且仅当 $\lceil x\rceil\leqslant n$。
    b) $n\leqslant x$ 当且仅当 $n\leqslant\lfloor x\rfloor$。
39. 若 x 为实数，求证 $\lfloor 2x\rfloor=\lfloor x\rfloor+\lfloor x+1/2\rfloor$。
40. 证明若 x 为实数，则 $\lfloor -x\rfloor=-\lceil x\rceil$，$\lceil -x\rceil=-\lfloor x\rfloor$。
41. 有些计算器上有个 INT 函数，当 x 为非负实数时 $\mathrm{INT}(x)=\lfloor x\rfloor$，当 x 为负实数时 $\mathrm{INT}(x)=\lceil x\rceil$。求证这一函数 INT 满足等式 $\mathrm{INT}(-x)=-\mathrm{INT}(x)$。
42. 令 a 和 b 为实数，且 $a<b$。用底函数及/或顶函数表示使 $a\leqslant n\leqslant b$ 的整数 n 的数目。
43. 令 a 和 b 的实数，且 $a<b$，用底函数及/或顶函数表示使 $a<n<b$ 的整数 n 的数目。
44. 含 n 个字位的数据需要用几个字节来编码？其中 n 是
    a) 4　　b) 10　　c) 500　　d) 3000
45. 含 n 个字位的数据需要用几个字节编码？其中 n 为
    a) 7　　b) 17　　c) 1001　　d) 28800
46. 用下面给出的传输率的连接在 10 秒钟内能传输多少 ATM 信元（参看例 23)？
    a) 每秒 128 千字位。
    b) 每秒 300 千字位。
    c) 每秒兆字位。
47. 数据在某以太网上以 1500 个八位字节为信息块传输。下面的数据量在这个以太网上各需要多少个信息块传输（注意字节就是 8 位字节)？
    a) 150 千字节的数据。
    b) 384 千字节的数据。
    c) 1.544 兆字节的数据。
    d) 45.3 兆字节的数据。
48. 画出从 $\mathbf{Z}$ 到 $\mathbf{Z}$ 的函数 $f(n)=1-n^2$ 的图像。
49. 画出从 $\mathbf{R}$ 到 $\mathbf{R}$ 的函数 $f(x)=\lfloor 2x\rfloor$ 的图像。

50. 画出从 **R** 到 **R** 的函数 $f(x)=\lfloor x/2\rfloor$ 的图像。
51. 画出从 **R** 到 **R** 的函数 $f(x)=\lfloor x\rfloor+\lfloor x/2\rfloor$ 的图像。
52. 画出从 **R** 到 **R** 的函数 $f(x)=\lceil x\rceil+\lfloor x/2\rfloor$ 的图像。
53. 画出下列各函数的图像：
    a) $f(x)=\lfloor x+1/2\rfloor$
    b) $f(x)=\lfloor 2x+1\rfloor$
    c) $f(x)=\lceil x/3\rceil$
    d) $f(x)=\lceil 1/x\rceil$
    e) $f(x)=\lceil x-2\rceil+\lfloor x+2\rfloor$
    f) $f(x)=\lfloor 2x\rfloor\lceil x/2\rceil$
    g) $f(x)=\lfloor\lceil x-1/2\rceil+1/2\rfloor$
54. 画出下列各函数的图像：
    a) $f(x)=\lceil 3x-2\rceil$
    b) $f(x)=\lceil 0.2x\rceil$
    c) $f(x)=\lfloor -1/x\rfloor$
    d) $f(x)=\lfloor x^2\rfloor$
    e) $f(x)=\lceil x/2\rceil\lfloor x/2\rfloor$
    f) $f(x)=\lfloor x/2\rfloor+\lceil x/2\rceil$
    g) $f(x)=\lfloor 2\lceil x/2\rceil+1/2\rfloor$
55. 求 $f(x)=x^3+1$ 的反函数。
56. 假定 f 是从 Y 到 Z 的可逆函数，g 是从 X 到 Y 的可逆函数。证明函数组合 $f\circ g$ 的反函数 $(f\circ g)^{-1}=g^{-1}\circ f^{-1}$。
57. 令 S 为全集 U 的子集。S 的特征函数 f_S 是从 U 到集合 $\{0, 1\}$ 的函数，使得若 x 属于 S 则 $f_S(x)=1$，若 x 不属于 S，则 $f_S(x)=0$。令 A，B 为集合。求证
    a) $f_{A\cap B}(x)=f_A(x)\cdot f_B(x)$
    b) $f_{A\cup B}(x)=f_A(x)+f_B(x)-f_A(x)\cdot f_B(x)$
    c) $f_{\overline{A}}(x)=1-f_A(x)$
    d) $f_{A\oplus B}(x)=f_A(x)+f_B(x)-2f_A(x)f_B(x)$
58. 假定 f 是从 A 到 B 的函数，A 和 B 为有限集，且 $|A|=|B|$。证明 f 是一对一的当且仅当它是映上的。

用来计算函数值的程序也许不能给出该函数在其所有元素上的正确值。例如程序可能因无限循环或内存溢出而不能给出某个正确值。为研究这种情况，我们用到**部分函数**的概念。从集合 A 到集合 B 的部分函数 f 是为 A 的一个子集的每个元素指定 B 中唯一值的指派。A 的这个子集称为 f 的定义区域。集合 A 和 B 分别称为 f 的**定义域**和**伴域**。我们说 f 在 A 中不属于定义区域的元素上未定义。我们用 $f: A\to B$ 表示 f 是从 A 到 B 的部分函数（这个符号与表示函数的符号一样。符号的上下文将决定 f 究竟是部分函数还是全函数）。当 f 的定义区域等于 A 时，就说 f 是**全函数**。

59. 对下列各个部分函数求它的定义域、伴域、定义区域及其未定义的值的集合。另外判断它是否为全函数。
    a) $f:\mathbf{Z}\to\mathbf{R}, f(n)=1/n$
    b) $f:\mathbf{Z}\to\mathbf{Z}, f(n)=\lceil n/2\rceil$
    c) $f:\mathbf{Z}\times\mathbf{Z}\to\mathbf{Q}, f(m,n)=m/n$
    d) $f:\mathbf{Z}\times\mathbf{Z}\to\mathbf{Z}, f(m,n)=mn$
    e) $f:\mathbf{Z}\times\mathbf{Z}\to\mathbf{Z}, f(m,n)=m-n$，如果 $m>n$

60. a) 证明从 A 到 B 的部分函数 f 可以看成从 A 到 $B\cup\{u\}$ 的函数 f^*，其中 u 不是 B 的元素，且

$$f^*(a)=\begin{cases} f(a), \text{若 } a \text{ 属于 } f \text{ 的定义区域} \\ u, \text{若 } f \text{ 在 } a \text{ 未定义} \end{cases}$$

b) 使用 a) 中的构造法，找出练习 59 中各部分函数对应的 f^*。

1.7 序列与求和

1.7.1 引言

序列用于表示元素的有序表。离散数学中以许多方式使用序列。我们在第 5 章将会看到序列用于表示某些计数问题的解。在计算机科学中序列也是一种重要的数据结构。本节回顾一下函数的概念，并介绍用于表示序列和序列各项之和的符号。

若无限集合中的元素可以列成表，就说该集合是可数的。我们将以对可数和不可数集合的讨论结束本节。

1.7.2 序列

序列是用于表示有序表的离散结构。

定义 1 序列是从整数集合的一个子集（通常是 $\{0,1,2,\cdots\}$ 或 $\{1,2,3,\cdots\}$）到某个集合 S 的函数。我们用 a_n 表示整数 n 的像，并称 a_n 为该序列的一个项。

我们用符号 $\{a_n\}$ 表示序列。(注意，a_n 表示序列 $\{a_n\}$ 的一个单项。还要注意，序列符号 $\{a_n\}$ 与集合符号是一样的。不过在使用这同一符号时，上下文会使它表示序列还是表示集合一目了然。)

我们按下标增加的次序列出序列中各项，以此来描述序列。

例 1 考虑序列 $\{a_n\}$，其中

$$a_n=1/n$$

从 a_1 开始，这一序列各项所成的表，即

$$a_1, a_2, a_3, a_4, \cdots,$$

开头几项是

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \cdots$$ ■

例 2 考虑序列 $\{b_n\}$，其中 $b_n=(-1)^n$。这一序列的各项的表 b_0，b_1，b_2，b_3，…的开头几项是

$$1,-1,1,-1,1,\cdots$$ ■

例 3 考虑序列 $c_n=5^n$。此序列各项的表 c_0，c_1，c_2，c_3，c_4，c_5，…的开头几项是

$$1,5,25,125,625,3125,\cdots$$ ■

计算机科学中常用的序列形式是

$$a_1, a_2, \cdots, a_n$$

这种有限的序列也称为串。这个串也写成 $a_1a_2\cdots a_n$（回忆位串，是字位的有限序列，在 1.1 节引入）。串 S 的长度是串中项的个数。空串是不含任何项的串，空串长度为 0。

例 4　$abcd$ 是长度为 4 的一个串。■

1.7.3　特殊的整数序列

离散数学中一个共同的问题是找出构造序列项的公式或通用规则。有时已知解题序列的几个项，目标是找出整个序列。尽管序列的开头几项并不能决定整个序列（其实初始几项相同的不同序列有无穷多个），但知道头几项可以帮助你做出对整个序列的合理猜测。一旦有了猜测，即可尝试证明你找出了正确的序列。

在尝试从几个初始项推导项的可能公式或规则时，试试找出这几个项的模式。也许可以看出能否找到从前面的那些项产生紧随其后的项的办法。可以问许多问题，其中较有用的一些是：

- 是否有同一个值的反复出现？
- 能否在前项上加一常量，或加一与项在序列中位置有关的量，就得到后项？
- 能否将前项乘以某个特定的量得出后项？
- 能否以某种方式组合前面的项以得到新项？

下面的几个例子解释了怎样解决这一类问题。

例 5　如果序列的头 10 个项是 1，2，2，3，3，3，4，4，4，4，求产生序列项的规则。

解　注意，整数 1 出现 1 次，整数 2 出现 2 次，整数 3 出现 3 次，而整数 4 出现 4 次。产生这一序列的一条合理规则是整数 n 恰出现 n 次。于是序列下面 5 项会全是 5，再后面的 6 项全是 6，等等。以这种方式产生的序列是一个可能的解序列。■

例 6　如果序列的头 10 项是5,11,17,23,29,35,41,47,53,59,产生序列项的规则是什么？

解　注意这头 10 项的每一项，除第一项外都是在第一项上加 6 得到的（注意到相邻两项的差是 6 即可看出这一点）。于是第 n 项可以 5 开始，再加上 $(n-1)$ 次 6 得到，也就是说，第 n 项是$5+6(n-1)=6n-1$。■

例 6 的解序列是算术序列，即形为 a，$a+d$，$a+2d$，$a+3d$，…，$a+nd$，…的序列。解序列中 $a=5$，$d=6$。

求解产生序列项的规则的另一有用技术是将问题中的序列项与熟知的某个整数序列的项比较，例如算术序列的项，几何序列（见例 12）的项，完全平方，完全立方，等等。表 1-20 列出了也许是你希望记住的某些序列的头 10 项。

例 7　如果序列$\{a_n\}$的前 10 项为1,7,25,79,241,727,2185,6559,19681,59047。猜一猜 a_n 的一个简单公式。

解　要解决这一问题，先看看相邻项的差，但找不出模式。在计算相邻项的比，看看每个项是否为前项的倍数时，发现这个比尽管不是常数，却接近 3。于是有理由怀疑这一序列

表 1-20 若干有用的序列

第 n 项	前 10 项
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, …
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, …
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, …
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, …
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, …
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, …

的项由含 3^n 的公式产生。将这一序列的项与序列 $\{3^n\}$ 的对应项比较，注意到第 n 项与 3 的幂的对应项少 2。由于对 $1\leqslant n\leqslant 10$，$a_n=3^n-2$，我们猜这一公式对所有 n 成立。■

本书大量离散数学内容中都出现整数序列。我们将会见到的序列包括素数序列（第 2 章），为 n 个离散对象排序的不同排序数（第 4 章），有名的 n 个盘子的汉诺塔难题求解需要的移动步数（第 5 章），n 个月之后岛上兔子的只数（第 5 章），以及为 n 个数排序需要进行的比较大小次数（第 8 章）。

除离散数学外，整数序列还惊人的出现在大量学科中，包括生物，物理，工程，化学和智力游戏。在过去 20 年中数学家 Neil Sloane㊀已经构造了 8000 多种不同的五花八门的整数序列，他还与 Simon Plouffe 合作出版了《整数序列百科全书》（[SIP195]）。网站上有个这种序列的扩展表供查阅，新的序列还会定期添加进去。网上还有一个程序供你从你提供的初始项从该百科全书中查找与之匹配的序列。

1.7.4 求和

下面引入求和符号。从用于表达求序列 $\{a_n\}$ 的项

$$a_m,\ a_{m+1},\ \cdots,\ a_n$$

之和的符号开始。我们用符号

$$\sum_{j=m}^{n} a_j$$

㊀ 斯朗（Neil Sloane，生于 1939 年） 斯朗依靠澳大利亚国家电话公司的助学金在墨尔本大学学习数学和电气工程。他掌握了与电话有关的许多工作技能，如在夏天打工时树电话杆。毕业后他设计了澳大利亚成本最低的电话网。1962 年他来到美国，在康奈尔大学学习电气工程。他的博士论文内容就是现在所说的神经网络。1969 年他接受了 Bell 实验室的工作，在许多领域干过，包括网络设计，编码理论，以及球体打包。他现在在 AT&T 实验室工作，是 1996 年 AT&T 从 Bell 实验室分裂出来时转到那里的。他最喜欢的问题之一是接吻问题（他创造的名字），该问题问的是在 n 维空间多少个 n 维球体可以组织在一起使它们全能与中央同样大小的一个 n 维球体接触？（2 维空间答案是 6，因为 6 个一分硬币可以摆成一圈且全都碰到中央的一分硬币。在 3 维空间答案是 12，12 个台球可以放在一起并且全都接触中央的一个台球。两个互相接触的台球称为“接吻”，于是就有了接吻问题和接吻次数等术语）。斯朗与 Andrew Odlyzko 证明了对 8 维和 24 维而言，最优的接吻次数分别是 240 和 196560。1，2，3，8 和 24 维的接吻次数是已知的，但其他维还是未知数。斯朗的著作中包括与 John Conway 合写的《*Sphere Packing, Lattices and Groups*》（球体打包、格和群）第 3 版；与 Jessie Mac Williams 合写的《*The Theory of Error – Dorrecting Codes*》（纠错码理论）；与 Simon Plouffe 合写的《*The Encyclopedia of Integer Sequences*》（整数序列百科全书）；以及与 Paul Nick 合写的《*The Rock Climbing Cuide to New Jersey Crags*》（新泽西州砂质泥灰岩攀岩指南）。最后一本书表现了他对攀岩的兴趣：在新泽西州有 50 多个攀岩场地。

表示 $a_m + a_{m+1} + \cdots + a_n$，其中变量 j 称为求和下标，而选用字母 j 作为变量名是随意的；换言之，我们可以用任何其他字母，如 i 或 k。或用符号表示：

$$\sum_{j=m}^{n} a_j = \sum_{i=m}^{n} a_i = \sum_{k=m}^{n} a_k$$

这里，求和下标从它的下界 m 开始到上界 n 终止，取遍其间所有整数值。大写的希腊字母 Σ 表示求和。我们举几个求和符号的例子。

例 8 表示序列 $\{a_n\}$ 前 100 项的和，其中 $a_n = 1/n$，$n = 1, 2, 3, \cdots$。

解 求和的下界是 1，上界是 100。于是这一和可写为

$$\sum_{j=1}^{100} j$$ ■

例 9 $\sum_{j=1}^{5} j^2$ 的值是什么？

解 $\sum_{j=1}^{5} j^2 = 1^2 + 2^2 + 3^2 + 4^3 + 5^2 = 1 + 4 + 9 + 16 + 25 = 55$ ■

例 10 $\sum_{k=4}^{8} (-1)^k$ 的值是什么？

解

$$\sum_{k=4}^{8} (-1)^k = (-1)^4 + (-1)^5 + (-1)^6 + (-1)^7 + (-1)^8$$
$$= 1 + (-1) + 1 + (-1) + 1 = 1$$ ■

有时需要对求和符号中的下标作移位。把两个求和符号相加而下标又不匹配时往往这样做。对下标移位时，必须对相应的加数做适当的变动。下面的例子解释了这一点。

例 11 假定已有用求和符号写的和

$$\sum_{j=1}^{5} j^2$$

但想要让下标从 0 到 4，而不是从 1 到 5。为此令 $k = j - 1$。这个新的求和下标是从 0 到 4 的，而项 j^2 变成了 $(k+1)^2$。因此

$$\sum_{j=1}^{5} j^2 = \sum_{k=0}^{4} (k+1)^2$$

很容易核实两者都是 $1 + 4 + 9 + 16 + 25 = 55$。 ■

例 12 几何序列是形如

$$a,\ ar,\ ar^2,\ ar^3,\ \cdots ar^k$$

的序列，其中初始项 a 和公比 r 均为实数。对几何序列项求和是常见的，这种求和称为几何级数。我们将求出几何序列前 $n+1$ 项的和 S 的公式，其中初项为 a，非零公比为 r，即求

$$S = \sum_{j=0}^{n} ar^j$$

要计算 S，首先在等式两边乘以 r，然后把得到的结果做如下处理：

$$rS = r\sum_{j=0}^{n} ar^j$$
$$= \sum_{j=0}^{n} ar^{j+1}$$

$$= \sum_{k=1}^{n+1} ar^k$$
$$= \sum_{k=0}^{n} ar^k + (r^{n+1} - a)\text{（此等式由求和下标移位 } k = j+1 \text{ 得到）}$$
$$= S + (ar^{n+1} - a)$$

从这些等式知

$$rS = S + (ar^{n+1} - a)$$

求 S 可得：若 $r \neq 1$

$$S = \frac{ar^{n+1} - a}{r - 1}$$

若 $r = 1$，显然 S 等于 $(n+1)a$。 ■

例 13 在许多情况下都要双重求和（例如分析计算机程序中的嵌套循环）。双重求和的例子是

$$\sum_{i=1}^{4} \sum_{j=1}^{3} ij$$

要计算双重和，首先扩开内层求和，然后再计算外层和：

$$\sum_{i=1}^{4} \sum_{j=1}^{3} ij = \sum_{i=1}^{4} (i + 2i + 3i)$$
$$= \sum_{i=1}^{4} 6i$$
$$= 6 + 12 + 18 + 24 = 60$$

可以用求和符号把函数的所有值加起来，也就是对下标集合的项求和，其中求和下标取集合的所有值。我们可以写下

$$\sum_{s \in S} f(s)$$

以表示函数值 $f(s)$ 对 S 中所有元素 s 求和。 ■

例 14 $\sum_{s \in \{0,2,4\}} s$ 的值是什么？

解 因为 $\sum_{s \in \{0,2,4\}} s$ 表示对集合 $\{0,2,4\}$ 中所有元素 s 的值的和，所以得到

$$\sum_{s \in \{0,2,4\}} s = 0 + 2 + 4 = 6$$ ■

某些和在离散数学中反复出现。掌握这些和的求和公式是有用的。表 1-21 提供了几个常出现的求和公式。

在例 12 中已经推导了该表中的第一个公式。其余的三个公式给出的是前 n 个正整数的和，它们的平方的和以及它们的立方的和。这三个公式可以用许多方式推导出来（例如参见本节末的练习 21 和 22）。还要注意，一旦知道了公式，就可以用数学归纳法很容易地证明它。数学归纳法是 3.2 节的内容。

例 15 说明了表 1-21 中的公式是很有用的。

例 15 求 $\sum_{k=50}^{100} k^2$。

表 1-21 几个有用的求和公式

求和	公式（闭形式）
$\sum_{k=0}^{n} ar^k$	$\frac{ar^{n+1}-a}{r-1}$，$r\neq 1$
$\sum_{k=1}^{n} k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^{n} k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^{n} k^3$	$\frac{n^2(n+1)^2}{4}$

解 首先注意$\sum_{k=1}^{100} k^2=\sum_{k=1}^{49} k^2+\sum_{k=50}^{100} k^2$，于是

$$\sum_{k=50}^{100} k^2=\sum_{k=1}^{100} k^2-\sum_{k=1}^{49} k^2$$

使用表 1-21 中的公式$\sum_{k=1}^{n} k^2=n(n+1)(2n+1)/6$，得

$$\sum_{k=50}^{100} k^2=\frac{100\cdot 101\cdot 201}{6}-\frac{49\cdot 50\cdot 99}{6}=338350-40425=297925$$ ■

1.7.5 基数（选读）

在 1.4 节已经定义了有限集合的基数，这就是该集合中元素的个数。用下面的定义可以把基数的概念扩展到所有集合，无论是有限还是无限集合。

定义 2 集合 A 和集 B 有同样的*基数*当且仅当存在从 A 到 B 的一一对应关系。

要看清这一定义与前面将有限集合的基数定义为该集合所含元素个数是一致的，请注意在两个含 n 个元素的有限集合中存在一一对应关系，这里 n 是非负整数。

现在我们把无限集合分成两组，一组是与自然数集合有相同基数的，另一组基数不同。

定义 3 有限集合或与自然数集合基数相同的集合都称为*可数的*。不是可数的集合称为*不可数的*。

我们举几个可数与不可数集合的例子。

例 16 求证奇正整数集合是可数集合。

解 为证明奇正整数集合是可数的，我们来建立这一集合与自然数集合之间的一一对应关系。考虑从自然数集合 $\mathbf{N}$ 到奇正整数集合的函数

$$f(n)=2n-1$$

我们证明这是个双向一对一的映上函数，从而 f 是一一对应关系。要证明 f 是一对一的，假定$f(n)=f(m)$。于是 $2n-1=2m-1$，所以 $n=m$；要证明 f 是映上的，假定 t 是个奇正整数。于是 t 比某个偶数 $2k$ 少 1，其中 k 是个自然数。从而$t=2k-1=f(k)$。图 1-18 中画出了这个一一对应关系。 ■

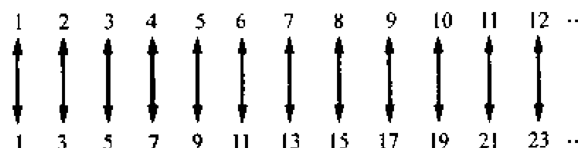


图 1-18 在 N 和奇正整数集合之间的一一对应

无限集合是可数的当且仅当可以把集合中的元素列成序列（以自然数为下标）。原因是从自然数集合到集合 S 的一一对应 f 可以用序列 a_1，a_2，…，a_n，…表示，其中 $a_1=f(1)$，$a_2=f(2),\cdots,a_n=f(n),\cdots$。例如，奇整数集合可以列成 a_1，a_2，…，a_n，…，其中 $a_n=2n-1$。

现在看一个不可数集合的例子。

例 17　证明实数集合是不可数集合。

解　要证明实数集合是不可数的，我们假定实数集合可数，并由此推出矛盾。若实数集合可数，那么在 0 与 1 之间的所有实数的子集也是可数的（因为可数集合的所有子集都是可数的；参看本节末练习 32）。在这一假设之下，0 和 1 之间的实数可以以某种顺序列出来，比如说 r_1，r_2，r_3，…。令这些实数的十进位表示为

$$r_1=0.d_{11}d_{12}d_{13}d_{14}\cdots$$
$$r_2=0.d_{21}d_{22}d_{23}d_{24}\cdots$$
$$r_3=0.d_{31}d_{32}d_{33}d_{34}\cdots$$
$$r_4=0.d_{41}d_{42}d_{43}d_{44}\cdots$$

其中 $d_{ij}\in\{0,1,2,3,4,5,6,7,8,9\}$。（例如，若 $r1=0.23794102...$，就有 $d_{11}=2$，$d_{12}=3$，$d_{13}=7$，等等。）于是可以产生一个新的实数，其十进位展开是 $r=0.d_1d_2d_3d_4...$ 其中十进数字由公式

$$d_i=\begin{cases}4, & 若\ d_{ii}\neq 4\\ 5, & 若\ d_{ii}=4\end{cases}$$

决定。做为一个例子，假定 $r_1=0.23794102\cdots$，$r_2=0.44590138\cdots$，$r_3=0.09118764\cdots$，$r_4=0.80553900\cdots$，等等。那么 $r=0.d_1d_2d_3d_4\cdots=0.4544...$，其中 $d_1=4$（因为 $d_{11}\neq 4$），$d_2=5$（因为 $d_{22}=4$），$d_3=4$（因为 $d_{33}\neq 4$），$d_4=4$（因为 $d_{44}\neq 4$），等等。

每个实数都有唯一的十进位展开（十进位展开中可能会出现从某一位起尾部全为 9，我们不算这种表示）。于是 r 不等于 r_1，r_2，…中的任何一个，因为，对每个 i，r 的十进展开与 r_i 的十进展开在小数点后的第 i 位不同。

由于在 0 和 1 之间有一个实数不在列出的序列中，0 和 1 之间所有实数可以列成序列的假定必定不成立。因此，0 和 1 之间所有实数不能列成序列，0 和 1 之间的实数集合不可数。任何含不可数子集的集合都是不可数的（参看本节末的练习 35）。所以实数集合是不可数的。■

练习

1．求序列 $\{a_n\}$ 的下列各项，其中 $a_n=2\cdot(-3)^n+5^n$。
　a）a_0　　b）a_1　　c）a_4　　d）a_5

2．序列 $\{a_n\}$ 的项 a_8 是什么？如果 a_n 为
　a）2^{n-1}　　b）7　　c）$1+(-1)^n$　　d）$-(-2)^n$

3．什么是序列 $\{a_n\}$ 的项 a_0,a_1,a_2 和 a_3？其中 a_n 为

a) 2^n+1　b) $(n+1)^{n+1}$　c) $\lfloor n/2\rfloor$　d) $\lfloor n/2\rfloor\lceil n/2\rceil$

4. 什么是序列 $\{a_n\}$ 的项 a_0, a_1, a_2 和 a_3？其中 a_n 为

a) $(-2)^n$　b) 3　c) $7+4^n$　d) $2^n+(-2)^n$

5. 列出下列各序列的前 10 项。

a) 从 2 开始的序列，它的后继项都比其前项多 3。

b) 序列中按增序列出各个正整数，每个数出现三次。

c) 序列中按增序列出各奇正整数，每个数出现两次。

d) 第 n 项为 $n!-2^n$ 的序列。

e) 从 3 开始的序列，它的后继项都是前项的两倍。

f) 头两项都是 1 的序列，它的后继项都是前面两项之和。（这是著名的斐波那契序列，我们将在后文中学习。）

g) 序列的第 n 项是数 n 的二进制位展开（在 2.3 节中定义）的字位数。

h) 序列的第 n 项是表示下标 n 的英文字中包含的字母个数。

6. 列出下列各序列的前 10 项：

a) 从 10 开始，以后各项都是从前一项减去 3 所得的序列。

b) 第 n 项为头 n 个正整数之和的序列。

c) 第 n 项为 3^n-2^n 的序列。

d) 第 n 项为 $\lfloor\sqrt{n}\rfloor$ 的序列。

e) 头两项为 1 和 2，每个后继项均为其前两项之和。

f) 序列的第 n 项为二进制展开（在 2.3 节定义）中有 n 个字位的最大整数。（用十进数写答案。）

g) 依下法顺序构造各项得的序列：从 1 开始，然后加上 1，然后乘以 1，再加上 2，再乘以 2，等等。

h) 序列的第 n 项是最大整数 k，使 $k!\leqslant n$。

7. 以 1,2,4 为起始项，找出至少三个不同的序列，它们的项均可用简单的公式或规则产生。

8. 以 3,5,7 为起始项，找出至少三个不同的序列，它们的项均可用简单的公式或规则产生。

9. 对下面的每列整数，给出一个简单的公式或规则，使它产生的整数序列的项就从给出的这列整数开始。

a) 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, ...

b) 1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, ...

c) 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, ...

d) 3, 6, 12, 24, 48, 96, 192, ...

e) 15, 8, 1, −6, −13, −20, −27, ...

f) 3, 5, 8, 12, 17, 23, 30, 38, 47, ...

g) 2, 16, 54, 128, 250, 432, 686, ...

h) 2, 3, 7, 25, 121, 721, 5041, 40321, ...

10. 对下面的每列整数，给出一个简单的公式或规则，使它产生的整数序列的项就从给出的这列整数开始。

a) 3, 6, 11, 18, 27, 38, 51, 66, 83, 102, ...

b) 7，11，15，19，23，27，31，35，39，43，...

c) 1，10，11，100，101，110，111，1000，1001，1010，1011，...

d) 1，2，2，2，3，3，3，3，3，5，5，5，5，5，5，5，...

e) 0，2，8，26，80，242，728，2186，6560，19682，...

f) 1，3，15，105，945，10395，135135，2027025，34459425，...

g) 1，0，0，1，1，1，0，0，0，0，1，1，1，1，1，...

h) 2，4，16，256，65536，4294967296，...

*11. 求证，若以 a_n 表示不是完全平方的第 n 个正整数，那么 $a_n = n + \{\sqrt{n}\}$，其中 $\{x\}$ 表示最接近 x 的整数。

*12. 令 a_n 表示序列1,2,2,3,3,3,4,4,4,4,5,5,5,5,5,6,6,6,6,6,6,...的第 n 项，这一序列是让整数 k 恰出现 k 次得到的。求证 $a_n = \lfloor\sqrt{2n} + 1/2\rfloor$。

13. 求下列各和的值。

a) $\sum_{k=1}^{5}(k+1)$　　b) $\sum_{j=0}^{4}(-2)^j$

c) $\sum_{i=1}^{10}3$　　d) $\sum_{j=0}^{8}(2^{j+1}-2^j)$

14. 令 S = {1，3，5，7}，求下列各和的值。

a) $\sum_{j\in S}j$　　b) $\sum_{j\in S}j^2$

c) $\sum_{j\in S}(1/j)$　　d) $\sum_{j\in S}1$

15. 求下列各几何级数项的和之值。

a) $\sum_{j=0}^{8}3\cdot 2^j$　　b) $\sum_{j=1}^{8}2^j$

c) $\sum_{j=2}^{8}(-3)^j$　　d) $\sum_{j=0}^{8}2\cdot(-3)^j$

16. 求下列各和之值。

a) $\sum_{j=0}^{8}(1+(-1)^j)$　　b) $\sum_{j=0}^{8}(3^j-2^j)$

c) $\sum_{j=0}^{8}(2\cdot 3^j+3\cdot 2^j)$　　d) $\sum_{j=0}^{8}(2^{j+1}-2^j)$

17. 计算下列各双重和。

a) $\sum_{i=1}^{2}\sum_{j=1}^{3}(i+j)$　　b) $\sum_{i=0}^{2}\sum_{j=0}^{3}(2i+3j)$

c) $\sum_{i=1}^{3}\sum_{j=0}^{2}i$　　d) $\sum_{i=1}^{2}\sum_{j=1}^{3}ij$

18. 计算下列各双重和。

a) $\sum_{i=1}^{3}\sum_{j=1}^{2}(i-j)$　　b) $\sum_{i=0}^{3}\sum_{j=0}^{2}(3i+2j)$

c) $\sum_{i=1}^{3}\sum_{j=0}^{3}j$　　d) $\sum_{i=0}^{2}\sum_{j=0}^{3}i^2j^3$

19. 求证 $\sum_{j=1}^{n}(a_j - a_{j-1}) = a_n - a_0$，其中 a_0，a_1，...，a_n 是个实数序列。这一类求和称为迭进。

20. 利用等式 $1/(k(k+1)) = 1/k - 1/(k+1)$ 及练习 19 中的公式计算 $\sum_{k=1}^{n}1/(k(k+1))$。

21. 从 $k=1$ 到 $k=n$ 在等式 $k^2-(k-1)^2=2k-1$ 两边求和，利用练习 19 的公式找出
    a) 计算 $\sum_{k=1}^{n}(2k-1)$ 的公式（头 n 个奇自然数之和）。
    b) 计算 $\sum_{k=1}^{n}k$ 的公式。

*22. 用练习 19 给出的技术，及练习 13 b）的结果，找出计算 $\sum_{k=1}^{n}k^2$ 的公式。

23. 求 $\sum_{k=100}^{200}k$（使用表 1-21）。

24. 求 $\sum_{k=99}^{200}k^3$（使用表 1-21）。

*25. 当 m 为正整数时求计算 $\sum_{k=0}^{m}\lfloor\sqrt{k}\rfloor$ 的公式。[提示：利用计算 $\sum_{k=1}^{n}k^2$ 的公式。]

*26. 当 m 为正整数时求计算 $\sum_{k=0}^{m}\lfloor\sqrt[3]{3}\rfloor$ 的公式。[提示：利用计算 $\sum_{k=1}^{n}k^3$ 的公式。]

还有一个用于乘积的符号。a_m，a_{m+1}，...，a_n 的乘积表示为

$$\prod_{j=m}^{n}a_j$$

27. 求下列乘积的值。
    a) $\prod_{i=0}^{10}i$　　b) $\prod_{i=5}^{8}i$
    c) $\prod_{i=1}^{100}(-1)^i$　　d) $\prod_{i=1}^{10}2$

阶乘函数在正整数 n 的值，用 $n!$ 表示，是从 1 到 n 的所有正整数的乘积。另外规定 $0!=1$。

28. 用乘积符号表示 $n!$。

29. 求 $\sum_{j=0}^{4}j!$。

30. 求 $\prod_{j=0}^{4}j!$。

31. 判断下列各集合是可数的还是不可数的。对可数的集合，给出自然数集合和该集合之间的一个一对一对应关系。
    a) 负整数。
    b) 偶整数。
    c) 0 和 1/2 之间的实数。
    d) 是 7 的倍数的整数。

*32. 判断下列各集合是否可数，对可数的集合，给出自然数集合和该集合之间的一个一对一对应。
    a) 不能被 3 整除的整数。
    b) 能被 5 整除但不能被 7 整除的整数。
    c) 十进制表示中只含数字 1 的实数。
    d) 十进制表示中只含数字 1 或 9 的实数。

33. 若 A 为不可数集合而 B 是可数的，$A-B$ 必定不可数吗？

34. 证明可数集合的子集也是可数的。

35. 证明若 A 是不可数的集合，且 $A\subseteq B$，则 B 也是不可数的。

36. 证明两个可数集合的并集也是可数的。

**37. 证明可数多个可数集合的并集也是可数的。

*38. 能写成两个整数之商的实数称为有理数。证明 0 和 1 之间的有理数是可数的。[提示：按 $p+q$ 增加的顺序列出这一集合中的元素，其中 p 和 q 分别是分数 p/q 的分子和分母，p/q 不可约分。]

*39. 证明所有位串的集合是可数的。

*40. 证明二次方程 $ax^2+bx+c=0$ 的解组成的实数集合是可数的，其中 a,b,c 为整数。

*41. 证明用特定的一种程序语言写的计算机程序组成的集合是可数的。[提示：以某种程序语言写的计算机程序可以想像成由有限字母集中字母组成的符号串。]

*42. 证明从正整数集合到集合 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ 的函数集合是不可数的。[提示：首先找一个 0 到 1 之间的实数集合到这些函数的一个子集的一对一对应关系。为此可以让实数 $0.d_1d_2\ldots d_n\ldots$ 对应函数 f，使 $f(n)=d_n$。]

*43. 如果有计算机程序能计算函数的值，就说这一函数是可计算的。利用练习 41 和 42 证明存在不可计算的函数。

1.8 函数增长

1.8.1 引言

假定一个计算机程序重新为 n 个整数的列表排序，使之依增序排列。此程序实用性的一个重要因素是计算机要化多长时间才能完成排序。分析的结果可能是，为 n 个整数（它们都小于某个指定的数）的列表重新排序需要的时间小于 $f(n)$ 微秒，其中 $f(n)=100n\log n+25n+9$。要分析此程序的实用性，我们需了解当 n 增长时，这一函数增长得多快。本节回顾某些用于估计函数增长的重要方法。我们将介绍最通用的分析函数增长的符号，即大 O 符号，我们将用这一符号引出若干关于函数增长的有用结论。

1.8.2 大 O 符号

常用一个专门的符号描述函数的增长。下面的定义给出这一符号。

定义 1 令 f 和 g 为从整数集合或实数集合到实数集合的函数。我们说 $f(x)$ 是 $O(g(x))$，如果有常数 C 和 k，使得只要 $x>k$，就有

$$|f(x)|\leqslant C|g(x)|$$

（$O(g(x))$ 读作 $f(x)$ 是大 O $g(x)$。）

注意 *要证明 $f(x)$ 是 $O(g(x))$，我们只需要找出一对常数 C 和 k，使得只要 $x>k$ 就有 $|f(x)|\leqslant C|g(x)|$。不过，符号定义要求的一对 C 和 k 不会是唯一的，而且只要有一对这样的数存在，就有无穷多个这样的数对。要看出这一点，只要注意若 C，k 是这样的一对，那么只要 C'，k' 满足 $C<C'$，$k<k'$，它们就也是一对，因为 $|f(x)|\leqslant C|g(x)|\leqslant C'|g(x)|$ 对所有满足 $x>k'>k$ 的 x 成立。*

例 1 证明 $f(x)=x^2+2x+1$ 是 $O(x^2)$。

解 因为只要 $x>1$，就有

$$0 \leqslant x^2+2x+1 \leqslant x^2+2x^2+x^2=4x^2$$

所以 $f(x)$ 是 $O(x^2)$。(要用大 O 的定义，取 $C=4$，$k=1$。这里不必用绝对值符号，因为当 x 为正数时，等式中所有函数都是正的。)

另一办法是注意到在 $x>2$ 时，$2x \leqslant x^2$，于是如果 $x>2$，我们有

$$0 \leqslant x^2+2x+1 \leqslant x^2+x^2+x^2=3x^2$$

(令 $C=3$ 及 $k=2$，再使用定义。)

请注意在 $f(x)$ 是 $O(x^2)$ 这一关系中，x^2 可以被函数值大于 x^2 的任何函数取代，例如 $f(x)$ 是 $O(x^3)$，$f(x)$ 是 $O(x^2+2x+7)$，等等。x^2 是 $O(x^2+2x+1)$ 也成立。因为只要 $x \geqslant 1$，$x^2 < x^2+2x+1$ 就成立。

图 1-19 为 x^2+2x+1 是 $O(x^2)$ 的图示。■

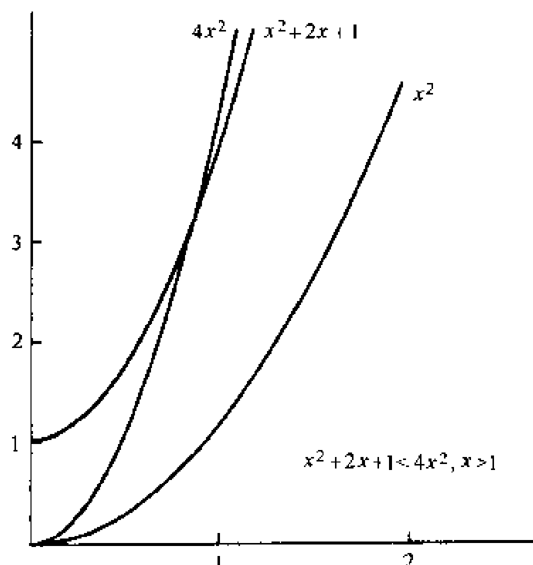


图 1-19 函数 x^2+2x+1 是 $O(x^2)$

注意在例 1 中有两个函数，$f(x)=x^2+2x+1$ 和 $g(x)=x^2$，使得 $f(x)$ 是 $O(g(x))$ 而且 $g(x)$ 是 $O(f(x))$。后一事实可以从不等式 $x^2 \leqslant x^2+2x+1$ 得到，其中 x 为任何非负实数。我们把满足上述两个大 O 关系的函数 $f(x)$ 和 $g(x)$ 称为同阶的。(参看 1.8.4 节。)

注意 $f(x)$ 是 $O(g(x))$ 的事实有时写作 $f(x)=O(g(x))$。不过这一写法中的等号并不代表真正的相等。其实这一记号说的是，对于这些函数定义域中足够大的数而言，函数 f 和 g 的值之间有个不等式成立。

大 O 符号已经在数学中使用了差不多一个世纪。在第 2 章我们会看到，这一记号也广泛应用于计算机科学的算法分析。德国数学家 Paul Bachmann[⊖] 1892 年在他的一本重要的数论书中首先引入了大 O 符号。大 O 符号有时又以德国数学家 Edmudn Landau（兰多）[⊖] 的名字称为兰多符号。兰多在他的工作中始终使用这一符号。大 O 符号在计算机科学中的普遍使用归功于 Donald Knuth（克努思）[⊖]，他还引入了将在本节随后介绍的大 Ω 和大 Θ 符号。

⊖ 巴赫曼(Paul Gustav Heinrich Bachmann, 1837－1920) 巴赫曼是路德教牧师的儿子，继承了他父亲虔诚的生活方式和对音乐的热爱。尽管巴赫曼早期的数学学习并不完全顺利，他的一位老师还是发现了他的数学才能。在瑞士从肺结核的疲痛中复原以后，巴赫曼学习了数学，首先在柏林大学，随后在哥庭根大学、在哥大他听了著名数论家狄利克雷（Dirichlet）的课。1862 年在德国数论家库默（Kummer）指导下获博士学位；他的论文内容是群论。巴赫曼曾先后担任布来斯劳（Breslau）大学和明斯特（MÜnster）大学教授，从教授位置上退休后，他继续数学写作、弹钢琴并且在报纸上作音乐评论。巴赫曼的数学论著包括 5 卷本对数论结果与方法的评述，2 卷本的初等数论，一本关于无理数的书和一本称为费马最后定理的著名猜测的书。在他 1892 年的书《*Analytische Zahlentheorie*》（解析数论）中引人了大 O 符号。

⊖ 兰多(Edmund Landau, 1877－1938)一位柏林妇科医生的儿子，在柏林完成高中和大学教育。1899 年在佛罗本尼乌斯（Frobenius）的指导下获得博士学位。兰多首先在柏林大学任教，后到哥庭根大学，他在那里任教授直到纳粹迫使他离开。兰多对数学的贡献主要在解析数论领域，特别是他对素数分布得到若干重要的结果。他完成了三卷本的数论评注及关于数论和数学分析的其他书籍。

若 $f(x)$ 是 $O(g(x))$，而对足够大的 x，$h(x)$ 是一个函数值的绝对值大于 $g(x)$ 的函数，则有 $f(x)$ 是 $O(h(x))$。换言之，在 $f(x)$ 是 $O(g(x))$ 这一关系中，函数 $g(x)$ 可以用具有更大绝对值的函数替换。要看清这一点，注意如果

$$|f(x)| \leqslant C|g(x)|, \text{ 只要 } x>k$$

且 $|h(x)|>|g(x)|$ 对所有 $x>k$ 成立，那么

$$|f(x)| \leqslant C|h(x)|, \text{ 只要 } x>k$$

于是 $f(x)$ 是 $O(h(x))$。

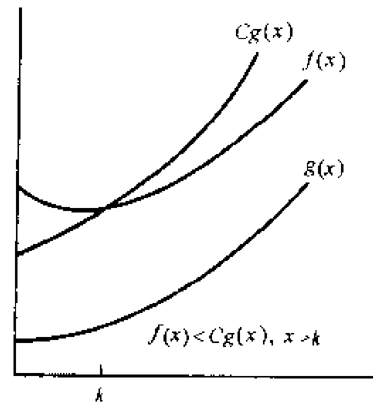


图 1-20 函数 $f(x)$ 是 $O(g(x))$

在使用大 O 符号时，在 $f(x)$ 是 $O(g(x))$ 这一关系中函数 g 总是选得尽可能小。（有时选自某个参考函数集合，例如形为 x^n 的函数，其中 n 为正整数。）

在随后的讨论中，我们差不多总是处理只有正值的函数。对这样的函数来说，在用大 O 对它作估计时可以不必涉及绝对值。图 1-20 是 $f(x)$ 是 $O(g(x))$ 的一个图解。

下面这个例子说明了怎样用大 O 符号来估计函数的增长。

例 2 证明 $7x^2$ 是 $O(x^3)$。

㊀ 克努思(Donald E. Knuth，1938 年生) 克努思在 Milwaukee 长大，他父亲在那里的路德教会高中教授簿记，并拥有一家小型的印刷厂。克努思是个优秀的学生，多次获得学业成就奖。他以反传统的方式运用才能，在八年级时赢了一次找字比赛，成绩是用 “Ziegler's Giant Bar” 中的字母组成了 4500 个单词。这次胜利为他的母校赢得一台电视机，为他班上的同学每人赢了一根糖棒。

克努思在 Case Institute of Technology（开思技术学院）就学时就放弃音乐而主修物理做了艰难的决策。然后他却又从物理转为数学，并于 1960 年获学士学位，由于教师们认为他的工作杰出，同时以特别奖的形式授予他硕士学位。在开思，他管理篮球队，并用他的才能发明了一个估价每位球员价值的公式。这一新奇的方法被《Newsweek》和 CBS 电视网上的 Walter Cronkite 报导。1960 年克努思开始在加利福尼亚理工学院做研究生，并于 1963 年获博士学位。在这一时间他还当顾问，为不同的计算机写编译程序。

克努思 1963 年加入了加利福尼亚理工学院的教师队伍，一直呆到 1968 年担任斯坦福大学教授。1992 年为集中精力写作他荣誉退休，保留教授头衔。他特别感兴趣的是为他的《计算机程序设计艺术》丛书完成新卷写作并更新旧卷。这套丛书是 1962 年他还是研究生时以编译程序为中心开始写作的，对计算机科学的发展产生了意义深远的影响。在常用的行话中，“克努思（Knuth）” 就意谓着 “计算机程序设计艺术”，也就意谓着数据结构和算法这一类问题的答案。

克努思是现代计算复杂性研究的奠基人。他对编译程序作出了基础性的贡献。对数学印刷的不满激发他发明了现在广泛使用的 TeX 和 Metafont 系统。TeX 已经成为计算机排印的一个标准语言。他获得众多奖项中的两项是 1974 年的图灵奖和卡特总统授给他的国家技术奖。

克努思为计算机科学和数学方面内容广泛的专业期刊写了许多文章。不过他的头一篇作品，1957 年还是一年级新生时写的，是对称为 “The Potrzebie Systems of Weights and Measures”（重量和长度的 Potrzebie 系统）的计量系统的模仿小品。该文首先登在《MAD 杂志》已经多次重印。他与他父亲一样是一位教堂风琴手。他还写作风琴乐曲。克努思相信，可以像写诗或作曲一样编写计算机程序。

对他书中的每个错误，克努思付 2.56 美元给第一个发现的人，对每个有意义的建议，付 0.32 美元。如果你寄给他一封信指出一个错误（你只能寄普通信件，因为他已放弃阅读电子邮件），他最终会通知你，你是否是第一个告诉他这一错误的人。要耐心等待，因为他收到的邮件太多。（作者寄给克努思一封报告错误的信，几年以后才收到回信，告诉我，我的报告比首先报告这一错误的信晚到了好几个月。）

解　只要 $x>7$，$7x^2<x^3$ 就成立。（只要在等式两边除以 x^2 就明白。）因此，在大 O 的定义中取 $C=1$ 而 $k=7$，得 $7x^2$ 是$O(x^3)$。■

例 3　例 2 证明了 $7x^2$ 是$O(x^3)$。x^3 也是 $O(7x^2)$吗？

解　要判定 x^3 是否$O(7x^2)$，必须判断是否存在常数 C 和k，使得只要 $x>k$，就有$x^3\leqslant C(7x^2)$。这一不等式等价于不等式 $x\leqslant 7C$，这是在上一不等式两边除以 x^2 得到的。由于 x 可以任意大，不可能存在这样的 C。于是 x^3 不是$O(7x^2)$。■

多项式常用于估计函数的增长，与其每当有多项式出现时都要分析多项式增长，不如找一个总是可以用来估计多项式增长的结论。下面的定理给出的就是这种结论。它证明了多项式的首项支配其增长，断定 n 阶或更低阶多项式是$O(x^n)$。

定理 1　令$f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$，其中 a_0，$a_1,\ldots$，a_{n-1}，a_n 为实数，那么$f(x)$是 $O(x^n)$。

证　用三角不等式，如果 $x>1$，我们有

$$\begin{aligned}|f(x)| &= |a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0|+\cdots+\\ &\leqslant |a_n|x^n+|a_{n-1}|x^{n-1}+\cdots+|a_1|x+|a_0|\\ &= x^n\ (|a_n|+|a_{n-1}|/x+\cdots+|a_1|/x^{n-1}+|a_0|/x^n)\\ &\leqslant x^n\ (|a_n|+|a_{n-1}|+\cdots+|a_1|+|a_0|)\end{aligned}$$

这说明，只要 $x>1$，就有

$$|f(x)|\leqslant Cx^n$$

其中 $C=|a_n|+|a_{n-1}|+\cdots+|a_1|+|a_0|$。于是$f(x)$是 $O(x^n)$。□

现在举几个与定义域为正整数集合的函数有关的例子。

例 4　怎样用大 O 符号估计前 n 个正整数的和？

解　由于前 n 个正整数都不超过 n，所以

$$1+2+\cdots+n\leqslant n+n+\cdots+n=n^2$$

由此不等式知 $1+2+\cdots+n$ 是$O(n^2)$，因为在大 O 定义中只须取 $C=1$ 和 $k=1$ 即可。（本例大 O 关系中的函数的定义域为正整数集合。）■

下面的例子是用大 O 估计阶乘函数和它的对数。这些估计对分析排序过程中使用的步数有重要作用。

例 5　给出阶乘函数和阶乘函数之对数的大 O 估计，其中阶乘函数 $f(n)=n!$ 的定义为：对正整数 n，

$$n!\ =1\cdot 2\cdot\cdots\cdot n$$

而 $0!\ =1$。例如，

$$1!=1,2!=1\cdot 2=2,3!=1\cdot 2\cdot 3=6,4!=1\cdot 2\cdot 3\cdot 4=24$$

注意函数 $n!$ 增长迅速。例如

$20! = 2\ 432\ 902\ 008\ 176\ 640\ 000$

解　只要注意到乘积中的每一项都不超过 n，就能得到 $n!$ 的大 O 估计，即

$$\begin{aligned} n! &= 1\cdot 2\cdot 3\cdot \cdots \cdot n \\ &\leqslant n\cdot n\cdot nD\cdot \cdots \cdot n \\ &= n^n \end{aligned}$$

这一不等式说明 $n!$ 是$O(n^n)$。对用于估计 $n!$ 的不等式两边同取对数，得

$$\log n! \leqslant \log n^n = n\log n$$

这表明 $\log n!$ 是$O(n\log n)$。 ■

例 6　在 3.2 节将证明对正整数 n，

$$n < 2^n$$

用这一不等式可以得知 n 是$O(2^n)$。(在大 O 的定义中取 $k = C = 1$。) 由于对数函数是增函数，只要在这一不等式两边取对数（以 2 为底），得

$$\log n < n$$

于是 $\log n$ 是$O(n)$。(仍取 $k = C = 1$。)

如果以 b 为底取对数，其中 b 不等于 2，就有 $\log_b n$ 是$O(n)$，因为

$$\log_b n = \frac{\log n}{\log b} < \frac{n}{\log b}$$

其中 n 是正整数。(我们使用了附录 1 定理 3 的公式 $\log_b n = \frac{\log n}{\log b}$。)

1.8.3　函数组合的增长

许多算法都由两个或更多独立的子过程组成，使用这样的算法用计算机求解带有一定大小输入的问题所需要的步数是这些过程使用的步数的和。想用大 O 估计需要的步数，必须找出对每个子过程所用步数的估计，再把这些估计组合在一起。

只要能把不同的大 O 估计组合在一起，就能提供对函数组合的大 O 估计。特别是往往要估计两个函数的和与积的增长。如果已知两个函数各自的大 O 估计，我们能得到什么结论呢？要弄清两个函数的和与积有什么样的估计，假定$f_1(x)$是 $O(g_1(x))$，$f_2(x)$是 $O(g_2(x))$。

从大 O 符号的定义，有常数 C_1，C_2，k_1，k_2，使得：当 $x > k_1$ 时

$$|f_1(x)| \leqslant C_1|g_1(x)|$$

当 $x > k_2$ 时

$$|f_2(x)| \leqslant C_2|g_2(x)|$$

要估计$f_1(x)$和 $f_2(x)$的和，注意

$$\begin{aligned} |(f_1 + f_2)(x)| &= |f_1(x) + f_2(x)| \\ &\leqslant |f_1(x)| + |f_2(x)| \quad (\text{用三角不等式} |a+b| \leqslant |a| + |b|) \end{aligned}$$

当 x 大于 k_1 和 k_2 时，从 $|f_1(x)|$ 和 $|f_2(x)|$ 的不等式得：

$$\begin{aligned}|f_1(x)|+|f_2(x)| &\leqslant C_1|g_1(x)|+C_2|g_2(x)| \\ &\leqslant C_1|g(x)|+C_2|g(x)| \\ &=(C_1+C_2)|g(x)| \\ &=C|g(x)|\end{aligned}$$

其中 $C=C_1+C_2$，而 $g(x)=\max(|g_1(x)|,|g_2(x)|)$。(这里 $\max(a,b)$ 表示最大值，亦即 a 和 b 中较大的一个。)

这一不等式说明 $|(f_1+f_2)(x)|\leqslant C\ |g(x)|$ 在 $x>k$ 时成立，其中 $k=\max\ (k_1,\ k_2)$。我们把这一有用的结果用下面的定理来描述。

定理 2 假定 $f_1(x)$ 是 $O(g_1(x))$，$f_2(x)$ 是 $O(g_2(x))$，那么 $(f_1+f_2)(x)$ 是 $O(\max(g_1(x),g_2(x)))$

对 f_1 和 f_2 的大 O 估计往往会使用同样的函数 g。在此情况下，由于 $\max(g_1(x),g_2(x))=g(x)$，定理 2 说明 $(f_1+f_2)(x)$ 也是 $O(g(x))$。下面的推论说的就是这一结果。

推论 假定 $f_1(x)$ 和 $f_2(x)$ 都是 $O(g(x))$，那么 $(f_1+f_2)(x)$ 也是 $O(g(x))$。

类似的方法可以推导出 f_1 和 f_2 的乘积的大 O 估计。当 x 大于 $\max\ (k_2,\ k_2)$ 时，我们有

$$\begin{aligned}|(f_1f_2)(x)| &= |f_1(x)||f_2(x)| \\ &\leqslant C_1|g_1(x)|C_2|g_2(x)| \\ &\leqslant C_1C_2|(g_1g_2)(x)| \\ &\leqslant C|(g_1g_2)(x)|\end{aligned}$$

其中 $C=C_1C_2$。从这一不等式知 $f_1(x)f_2(x)$ 是 $O(g_1g_2)$，因为存在常数 C 和 k，即 $C=C_1C_2$，$k=\max(k_1,k_2)$，使得只要 $x>k$，$|(f_1f_2)(x)|\leqslant C|g_1(x)g_2(x)|$。下面的定理说的即是这一结果。

定理 3 假定 $f_1(x)$ 是 $O(g_1(x))$，$f_2(x)$ 是 $O(g_2(x))$，那么 $(f_1f_2)(x)$ 是 $O(g_1(x)g_2(x))$。

用大 O 符号来估计函数的目的，是选一个相对增长较慢的函数 $g(x)$，使得 $f(x)$ 是 $O(g(x))$。下面的例子说明怎样利用定理 2 和定理 3 来实现这一目标。这些例子中给出的这一类分析常用于分析用计算机程序解题时用的时间。

例 7 给出 $f(n)=3n\log(n!)+(n^2+3)\log n$ 的大 O 估计，其中 n 是一个正整数。

解 首先估计乘积 $3n\log(n!)$。从例 5 我们知道 $\log\ (n!)$ 是 $O(n\log n)$。由这一估计及 $3n$ 是 $O(n)$ 的事实，定理 3 给出的估计为 $3n\log\ (n!)$ 是 $O(n^2\log n)$。

下一步估计乘积 $(n^2+3)\log n$。由于 $(n^2+3)<2n^2$ 在 $n>2$ 时成立，n^2+3 是 $O(n^2)$。因此由定理 3 知 $(n^2+3)\log n$ 是 $O(n^2\log n)$。用定理 2 把两个乘积的估计组合在一起得 $f(n)=3n\log(n!)+(n^2+3)\log n$ 是 $O(n^2\log n)$。

例 8 给出$f(x)=(x+1)\log(x^2+1)+3x^2$的大 O 估计。

解 首先找$(x+1)\log(x^2+1)$的大 O 估计。注意 $(x+1)$ 是$O(x)$。再由 $x>1$ 时 $x^2+1\leqslant 2x^2$，知 $x>2$ 时

$$\log(x^2+1)\leqslant\log(2x^2)=\log2+\log x^2=\log2+2\log x\leqslant 3\log x$$

这说明$\log(x^2+1)$是 $O(\log x)$。

从定理 3 知$(x+1)\log(x^2+1)$是 $O(x\log x)$。由于$3x^2$是$O(x^2)$，定理 2 告诉我们$f(x)$是 $O(\max(x\log x,x^2))$。因为 $x>1$ 时，$x\log x\leqslant x^2$，于是$f(x)$是 $O(x^2)$。■

前面提过，大 O 符号常用于估计一个特定的计算机过程或算法解题时需要的操作步数。做这些估计时常用的函数包括：

$$1,\ \log n,\ n,\ n\log n,\ n^2,\ 2^n,\ n!$$

用微积分可以证明，上列函数中每个函数都小于表中列在它后面的函数，准确地说是每个函数与它后面的函数的比在 n 无限增长时趋向于0。图 1-21 画出了这些函数的图像，图中函数值的每个刻度都是它前面的刻度的两倍。

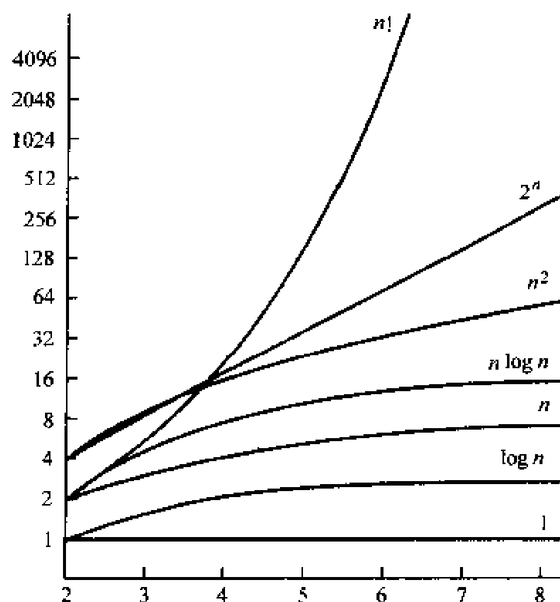


图 1-21 大 O 估计常用函数增长图

1.8.4 大 Ω 和大 Θ 符号

大 O 符号广泛用于描述函数的增长，但它有局限性，特别是当$f(x)$是$O(g(x))$时，我们只有用$g(x)$估计 $f(x)$的大小对大 x 值的一个上限。大 O 符号不能提供对大的x 值$f(x)$之大小的下限。为此我们使用大 Ω 符号。当我们希望给出函数$f(x)$的规模相对于参照函数$g(x)$的上限和下限时，我们使用大 Θ 符号。大 Ω 和大 Θ 符号都是克努思在 20 世纪 70 年代引入的。他引入这两个符号的动机是纠正人们需要函数的上限和下限时对大 O 符号的误用。

现在我们定义大 Ω 符号并解释怎样使用它。然后再定义大 Θ 并解释其使用。

在大 O 和大 Ω 符号之间有很强的联系。特别是$f(x)$是$\Omega(g(x))$当且仅当$g(x)$是$O(f(x))$。我们把这一事实的证明作为练习留给读者。

定义 2 令 f 和 g 为从整数集合或实数集合到实数集合的函数，我们说$f(x)$是$\Omega(g(x))$，如果存在正常数 C 和k，使得在 $x>k$ 时

$$|f(x)|\geqslant C|g(x)|$$

(读作$f(x)$是大Ω$g(x)$。)

例 9 函数$f(x)=8x^3+5x^2+7$是$\Omega(g(x))$，其中$g(x)=x^3$。由于$f(x)=8x^3+5x^2+7\geqslant 8x^3$对所有正实数都成立，所以上述说法成立。它等价于$g(x)=x^3$是 $O(8x^3+5x^2+7)$，而只须把不等式颠倒过来写直接就得到这一结论。■

通常重要的是知道相对于一个简单参照函数而言某函数增长的阶。简单函数指的是如 x^n（其中 n 是正整数）或 c^x（其中 $c>1$）这样的函数。要知道函数增长的阶，就需要了解函数大小的上界和下界。也就是说，给定一个函数 $f(x)$，我们想要一个参照函数 $g(x)$，使得 $f(x)$ 是 $O(g(x))$ 和 $f(x)$ 是 $\Omega(g(x))$。下面定义的大 Θ 符号表达这两个关系，即提供函数大小的上界，又提供函数大小的下界。

定义 3　令 f 和 g 为从整数集合或实数集合到实数集合的函数。如果 $f(x)$ 是 $O(g(x))$ 及 $f(x)$ 是 $\Omega(g(x))$，就说 $f(x)$ 是 $\Theta(g(x))$。若 $f(x)$ 是 $\Theta(g(x))$，就说"$f(x)$ 是大 $\Theta g(x)$"，也说 $f(x)$ 是 $g(x)$ 阶的。

若 $f(x)$ 是 $\Theta(g(x))$，$g(x)$ 也是 $\Theta(f(x))$。在使用大 Θ 符号时，$\Theta(g(x))$ 中的 $g(x)$ 通常是一个相对简单的参照函数，如 x^n，c^x，$\log x$ 等等，而 $f(x)$ 可能会相对复杂些。

例 10　我们已证明（例 4）前 n 个正整数的和是 $O(n^2)$。这个和是 n^2 阶的吗？

解　令 $f(n)=1+2+3+\cdots+n$，由于已知 $f(n)=O(n^2)$，为证明 $f(n)$ 是 n^2 阶的，只需找到正整数 C 使得对足够大的 n，$f(n)>Cn^2$。为获得这一和的下界，我们可以丢掉这些项里面的前一半。只把大于 $\lceil n/2\rceil$ 的项加起来，得

$$\begin{aligned}1+2+\cdots+n &\geq \lceil n/2\rceil+(\lceil n/2\rceil+1)+\cdots+n\\ &\geq \lceil n/2\rceil+\lceil n/2\rceil+\cdots+\lceil n/2\rceil\\ &=(n-\lceil n/2\rceil+1)\lceil n/2\rceil\\ &\geq (n/2)(n/2)\\ &=n^2/4\end{aligned}$$

这说明 $f(n)$ 是 $\Omega(n^2)$。我们得出结论：$f(n)$ 是 n^2 阶的。用符号表示，$f(n)$ 是 $\Theta(n^2)$。■

可以证明，如果能找到正实数 C_1，C_2 和正实数 k，使得对 $x\geq k$，

$$C_1|g(x)|\leq|f(x)|\leq C_2|g(x)|$$

则 $f(x)$ 是 $\Theta(g(x))$，因为上式说明 $f(x)$ 是 $O(g(x))$ 及 $f(x)$ 是 $\Omega(g(x))$。

例 11　证明 $3x^2+8x\log x$ 是 $\Theta(x^2)$。

解　因为 $0\leq 8x\log x\leq 8x^2$，所以对 $x\geq 1$，$3x^2+8x\log x\leq 11x^2$。从而 $3x^2+8x\log x$ 是 $O(x^2)$。显然 x^2 是 $O(3x^2+8x\log x)$，也就是 $3x^2+8x\log x$ 是 $\Theta(x^2)$。■

一个有用的事实是，多项式的首项决定它的阶。例如，若 $f(x)=3x^5+x^4+17x^3+2$，那么 $f(x)$ 是 x^5 阶的。下面的定理给出的就是这一事实，其证明留在本节末的练习中由读者来完成。

定理 4　令 $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$，其中 a_0，a_1，…，a_n 为实数，且 $a_n\neq 0$。则 $f(x)$ 是 x^n 阶的。

例 12　多项式 $3x^8+10x^7+221x^2+1444$，$x^{19}-18x^4-10112$ 和 $-x^{99}+40001x^{98}+100003x$ 分别是 x^8，x^{19} 和 x^{99} 阶的。■

不幸的是，正如克努思已观察到的，大 O 符号常被粗心的作者和演讲者误用，以为其

含义与大Θ相同。当你见到有人使用大O符号时，别忘了它常被误用。近来的倾向是，如果需要函数大小的上界和下界，就使用大Θ。

练习

1. 判断下列各函数是否为$O(x)$。
   a) $f(x)=10$　　b) $f(x)=3x+7$
   c) $f(x)=x^2+x+1$　　d) $f(x)=5\log x$
   e) $f(x)=\lfloor x\rfloor$　　f) $f(x)=\lceil x/2\rceil$
2. 判断下列各函数是否为$O(x^2)$。
   a) $f(x)=17x+11$　　b) $f(x)=x^2+1000$
   c) $f(x)=x\log x$　　d) $f(x)=x^4/2$
   e) $f(x)=2^x$　　f) $f(x)=\lfloor x\rfloor\cdot\lceil x\rceil$
3. 用$f(x)$是$O(g(x))$这一事实的定义证明x^4+9x^3+4x+7是$O(x^4)$。
4. 用$f(x)$是$O(g(x))$这一事实的定义证明2^x+17是$O(3^x)$。
5. 求证$(x^2+1)/(x+1)$是$O(x)$。
6. 求证$(x^3+2x)/(2x+1)$是$O(x^2)$。
7. 对下列每个函数求最小的整数n使$f(x)$是$O(x^n)$。
   a) $f(x)=2x^3+x^2\log x$
   b) $f(x)=3x^3+(\log x)^4$
   c) $f(x)=(x^4+x^2+1)/(x^3+1)$
   d) $f(x)=(x^4+5\log x)/(x^4+1)$
8. 对下列每个函数求最小的整数n使$f(x)$是$O(x^n)$。
   a) $f(x)=2x^2+x^3\log x$
   b) $f(x)=3x^5+(\log x)^4$
   c) $f(x)=(x^4+x^2+1)/(x^4+1)$
   d) $f(x)=(x^3+5\log x)/(x^4+1)$
9. 求证$x^2+4x+17$是$O(x^3)$，但x^3不是$O(x^2+4x+17)$。
10. 求证x^3是$O(x^4)$，但x^4不是$O(x^3)$。
11. 求证$3x^4+1$是$O(x^4/2)$，而且$x^4/2$是$O(3x^4+1)$。
12. 求证$x\log x$是$O(x^2)$但x^2不是$O(x\log x)$。
13. 求证2^n是$O(3^n)$但3^n不是$O(2^n)$。
14. 若$g(x)$是下面给出的函数，x^3是$O(g(x))$吗？[例如，如果$g(x)=x+1$，这一问题问的是x^3是$O(x+1)$吗？]
   a) $g(x)=x^2$　　b) $g(x)=x^3$
   c) $g(x)=x^2+x^3$　　d) $g(x)=x^2+x^4$
   e) $g(x)=3^x$　　f) $g(x)=x^3/2$
15. 解释一个函数是$O(1)$的含义。
16. 求证若$f(x)$是$O(x)$，那么$f(x)$是$O(x^2)$。
17. 假定$f(x)$，$g(x)$和$h(x)$为函数，使$f(x)$是$O(g(x))$，$g(x)$是$O(h(x))$。证明

$f(x)$是 $O(h(x))$。

18. 令 k 为正整数，证明 $1^k+2^k+\cdots+n^k$ 是 $O(n^{k+1})$。
19. 对下列各函数给出一个尽可能好的大 O 估计。
    a) $(n^2+8)(n+1)$
    b) $(n\log n+n^2)(n^3+2)$
    c) $(n!\ +2^n)(n^3+\log(n^2+1))$
20. 给下列各函数大 O 估计。在估计$f(x)$是$O(g(x))$中，要使用阶最小的一个简单函数$g(x)$。
    a) $(n^3+n^2\log n)(\log n+1)+(17\log n+19)(n^3+2)$
    b) $(2^n+n^2)(n^3+3^n)$
    c) $(n^n+n2^n+5^n)(n!+5^n)$
21. 给下列各函数一个大 O 估计。在估计$f(x)$是$O(g(x))$中，要使用阶最小的一个简单函数$g(x)$。
    a) $n\log(n^2+1)+n^2\log n$
    b) $(n\log n+1)^2+(\log n+1)(n^2+1)$
    c) $n^{2^n}+n^{n^2}$
22. 对练习 1 中的各函数，判断它是否为$\Omega(x)$和 $\Theta(x)$。
23. 对练习 2 中的各函数，判断它是否为$\Omega(x^2)$和 $\Theta(x^2)$。
24. a) 证明 $3x+7$ 是$\Theta(x)$。
    b) 证明 $2x^2+x-7$ 是$\Theta(x^2)$。
    c) 证明$\lfloor x+1/2\rfloor$是$\Theta(x)$。
    d) 证明$\log(x^2+1)$是$\Theta(\log_2 x)$。
    e) 证明 $\log_{10}x$ 是$\Theta(\log_2 x)$。
25. 证明$f(x)$是 $\Theta(g(x))$当且仅当$f(x)$是 $O(g(x))$且$g(x)$是 $O(f(x))$。
26. 若$f(x)$和 $g(x)$是从实数集到实数集的函数，求证$f(x)$是 $O(g(x))$当且仅当$g(x)$是 $\Omega(f(x))$。
27. 若$f(x)$和 $g(x)$为从实数集到实数集的函数，求证$f(x)$是 $\Theta(g(x))$当且仅当存在正常数 k，C_1 和 C_2 使得。只要 $x>k$，就有 $C_1|g(x)|\leqslant|f(x)|\leqslant C_2|g(x)|$。
28. a) 找出练习 27 中要求的 k，C_1，C_2，由此证明 $3x^2+x+1$ 是$\Theta(3x^2)$。
    b) 用图形表示 a) 中的关系，即给出函数$3x^2+x+1, C_1 3x^2$ 和 $C_2\cdot 3x^2$ 的图形，在 x－轴上标出 k，其中 k，C_1，C_2 是 a) 中你找到的证明 $3x^2+x+1$ 是$\Theta(3x^2)$的常数。
29. 用图形表示$f(x)$是 $\Theta(g(x))$这一关系。画出$f(x)$，$C_1|g(x)|$，$C_2|g(x)|$的图形并在 x 轴上标出常数k。
30. 解释函数为$\Omega(1)$的含义。
31. 解释函数为$\Theta(1)$的含义。
32. 给出头 n 个奇正整数的乘积的一个大O 估计。
33. 若 f 和g 为实数值函数，使$f(x)$是 $O(g(x))$，求证$f^k(x)$是 $O(g^k(x))$。
    [注意$f^k(x)=f(x)^k$ 。]

34. 若$f(x)$是$O(\log_b x)$，其中$b>1$，求证$f(x)$是$O(\log_a x)$，其中$a>1$。
35. 若$f(x)$是$O(g(x))$，其中f和g是无限增长的函数，求证$\log|f(x)|$是$O(\log|g(x)|)$。
36. 假定$f(x)$是$O(g(x))$。能否推断$2^{f(x)}$是$O(2^{g(x)})$?
37. 令$f_1(x)$和$f_2(x)$为从实数集合到正实数集合的函数。求证：若$f_1(x)$和$f_2(x)$均为$\Theta(g(x))$，其中$g(x)$是从实数集合到正实数集合的一个函数，则$f_1(x)+f_2(x)$是$\Theta(g(x))$。如果$f_1(x)$能取负值。这一结论还成立吗?
38. 假定$f(x)$，$g(x)$和$h(x)$是函数，使$f(x)$是$\Theta(g(x))$，$g(x)$是$\Theta(h(x))$。求证$f(x)$是$\Theta(h(x))$。
39. 若$f_1(x)$,$f_2(x)$为从正整数集合到正实数集合的函数，且$f_1(x)$和$f_2(x)$都是$\Theta(g(x))$。$(f_1-f_2)(x)$是否也是$\Theta(g(x))$? 证明它成立或给出一个反例。
40. 若$f_1(x)$和$f_2(x)$为从正整数集合到实数集合的函数，且$f_1(x)$是$\Theta(g_1(x))$，$f_2(x)$是$\Theta(g_2(x))$。求证$(f_1f_2)(x)$是$\Theta(g_1g_2(x))$。
41. 找从正整数集合到实数集合的函数$f(x)$和$g(x)$，使$f(n)$不是$O(g(n))$，$g(n)$也不是$O(f(n))$。
42. 用图形表示$f(x)$是$\Omega(g(x))$的关系。画出函数$f(x)$和$Cg(x)$，同时在x轴上标出常数k。
43. 若$f_1(x)$是$\Theta(g_1(x))$，$f_2(x)$是$\Theta(g_2(x))$，且对所有实数$x>0$，$f_2(x)\neq 0$，$g_2(x)\neq 0$，求证$(f_1/f_2)(x)$是$\Theta((g_1/g_2)(x))$。
44. 求证：若$f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$，其中$a_0$，$a_1$，$\cdots$，$a_n$为实数，且$a_n\neq 0$，则$f(x)$是$\Theta(x^n)$。

大O，大Θ和大Ω等符号可以推广到多元函数。例如，语句$f(x,y)$是$O(g(x,y))$的含义是：存在常数C，k_1和k_2，使得对$x>k_1$和$y>k_2$，$|f(x,y)|\leqslant C|g(x,y)|$。

45. 定义命题$f(x,y)$是$\Theta(g(x,y))$。
46. 定义命题$f(x,y)$是$\Omega(g(x,y))$。
47. 证明$(x^2+xy+x\log y)^3$是$O(x^6,y^3)$。
48. 证明$x^5y^3+x^4y^4+x^3y^5$是$\Omega(x^3y^3)$。
49. 证明$\lfloor xy\rfloor$是$O(xy)$。
50. 证明$\lceil xy\rceil$是$\Omega(xy)$。

以下的问题涉及另一类渐近符号，称为小o符号，由于小o符号以极限概念为基础，对那些问题微积分知识是必要的。我们说$f(x)$是$o(g(x))$[读作$f(x)$是小o $g(x)$]，如果

$$\lim_{x\to\infty}\frac{f(x)}{g(x)}=0$$

51. (需要微积分) 求证
    a) x^2是$o(x^3)$　　b) $x\log x$是$o(x^2)$
    c) x^2是$o(2^x)$　　d) x^2+x+1不是$o(x^2)$
52. (需要微积分)
    a) 求证：若函数$f(x)$和$g(x)$使得$f(x)$是$o(g(x))$，c为常数，则$(cf)(x)$是$o(g(x))$，其中$(cf)(x)=cf(x)$。

b) 求证：若 $f_1(x)$，$f_2(x)$ 和 $g(x)$ 为函数，使得 $f_1(x)$ 是 $o(g(x))$，$f_2(x)$ 是 $o(g(x))$，则 $(f_1+f_2)(x)$ 是 $o(g(x))$，其中 $(f_1+f_2)(x)=f_1(x)+f_2(x)$。

53. (需要微积分) 画出 $x\log x$，x^2 及 $x\log x/x^2$ 的图形以表示 $x\log x$ 是 $o(x^2)$。解释这一图形怎样说明 $x\log x$ 是 $o(x^2)$。

54. (需要微积分) 用图形表示 $f(x)$ 是 $o(g(x))$。画出 $f(x)$，$g(x)$ 和 $f(x)/g(x)$ 的图形。

*55. (需要微积分) 假定 $f(x)$ 是 $o(g(x))$。能是否由此推出 $2^{f(x)}$ 是 $o(2^{g(x)})$？

*56. (需要微积分) 假定 $f(x)$ 是 $o(g(x))$。能否由此推出 $\log|f(x)|$ 是 $o(\log|g(x)|)$？

57. (需要微积分) 本练习中的两部分描述了小 o 和大 O 符号之间的关系。

a) 求证：若函数 $f(x)$ 和 $g(x)$ 使 $f(x)$ 是 $o(g(x))$，则 $f(x)$ 是 $O(g(x))$

b) 求证：若函数 $f(x)$ 和 $g(x)$ 使 $f(x)$ 是 $O(g(x))$，那么不一定能推出 $f(x)$ 是 $o(g(x))$。

58. (需要微积分) 求证：若 $f(x)$ 是 n 阶多项式，$g(x)$ 是 m 阶多项式，且 $m>n$，则 $f(x)$ 是 $o(g(x))$。

59. (需要微积分) 求证：若 $f_1(x)$ 是 $o(g(x))$，$f_2(x)$ 是 $o(g(x))$，那么 $f_1(x)+f_2(x)$ 是 $o(g(x))$

60. (需要微积分) 令 H_n 为第 n 调和数

$$H_n = 1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}$$

求证：H_n 是 $O(\log n)$。[提示：建立不等式

$$\sum_{j=2}^{n}\frac{1}{j} < \int_1^n \frac{1}{x}dx$$

办法是证明对 $j=2, 3, \cdots, n$，以 $j-1$ 到 j 为底，$\frac{1}{j}$ 为高的所有长方形的面积之和小于曲线 $y=1/x$ 下面从 2 到 n 的面积。]

*61. 求证 $n\log n$ 是 $O(\log n!)$。

62. 判断 $\log(n!)$ 是 $O(n\log n)$ 是否成立，给出充分理由。

令 $f(x)$ 和 $g(x)$ 为从实数集合到实数集合的函数。如果 $\lim_{x\to\infty} f(x)/g(x)=1$，就说 $f(x)$ 和 $g(x)$ 渐近，写作 $f(x)\sim g(x)$。

63. (需要微积分) 对下列每对函数，判断 f 和 g 是否渐近。

a) $f(x)=x^2+3x+7, g(x)=x^2+10$

b) $f(x)=x^2\log x, g(x)=x^3$

c) $f(x)=x^4+\log(3x^8+7), g(x)=(x^2+17x+3)^2$

d) $f(x)=(x^3+x^2+x+1)^4, g(x)=(x^4+x^3+x^2+x+1)^3$

e) $f(x)=\log(x^2+1), g(x)=\log x$

f) $f(x)=2^{x+3}, g(x)=2^{x+7}$

g) $f(x)=2^{2^x}, g(x)=2^{x^2}$

关键术语和结果

逻辑（1.1－1.3 节）

术语

命题：一个或成真或为假的语句

真值：真或假

$\neg p$（p 的否定）：与 p 的真值相反的命题

逻辑运算符：用于组合命题的运算符

复合命题：用逻辑运算符组合命题构造出的命题

真值表：显示命题真值的表

$p \vee q$（p 和 q 的析取）：除非 p 和 q 均为假，否则为真的命题

$p \wedge q$（p 和 q 的合取）：只有在 p 和 q 均为真时才为真的命题

$p \oplus q$（p 和 q 的异或）：当 p 和 q 中恰有一个为真时才为真的命题

$p \rightarrow q$（p 蕴含 q）：只有在 p 为真而 q 为假时才为假的命题

$p \rightarrow q$ 的反蕴含：蕴含关系 $q \rightarrow p$

$p \rightarrow q$ 的倒置蕴含：蕴含关系 $\neg q \rightarrow \neg p$

$p \leftrightarrow q$（双向蕴含）：只有 p 和 q 真值相同时才为真的命题

字位：0 或 1

布尔变量：以 0 或 1 为值的变量

字位运算：一个或多个字位上的运算

位串：一串字位

按位运算：位串上的运算，对一个位串的字位和另一位串的对应字位作运算

永真式：永远为真的复合命题

矛盾：永远为假的复合命题

可能式：有时成真有时为假的复合命题

逻辑等价：复合命题是逻辑等价的，如果它们总有同样的真值

命题函数：谓词和变量的组合

论域：命题函数中变量的定义域

$\exists x p(x)$（$p(x)$ 的存在量化）：当且仅当在论域中存在一个 x 使 $p(x)$ 为真时才为真的命题

$\forall x p(x)$（$p(x)$ 的全称量化）：当且仅当论域中的所有 x 值 $p(x)$ 均为真时才为真的命题

自由变量：命题函数中未绑定的变量

结果　1.2 节表 1-12 和表 1-13 给出逻辑等价关系。

集合（1.4－1.5 节）

术语

集合：一组互不相同的对象

公理：理论的一个基本假设

悖论：逻辑上的不一致性

集合的元素，成员：集合中的一个对象

$\emptyset$（空集）：没有成员的集合

全集：包含考虑中的所有对象的集合

文氏图：一个或多个集合的一种图形表示

$S=T$（集合相等）：S 和 T 有相同的元素

$S\subseteq T$（S 是 T 的子集）：S 的每个元素也是 T 的元素

$S\subset T$（S 是 T 的真子集）：S 是 T 的子集，且 $S\neq T$

有限集：含 n 个元素的集合，其中 n 是非负整数

无限集：不是有限集的集合

$|S|$（S 的基数）：S 中元素的个数

$P(S)$（S 的幂集合）：S 的所有子集的集合

$A\cup B$（A 和 B 的并集）：包含那些至少属于 A 和 B 之一的元素的集合

$A\cap B$（A 和 B 的交集）：包含那些既属于 A 又属于 B 的元素的集合

$A-B$（A 和 B 的差集）：包含那些只属于 A 而不属于 B 的元素的集合

$\overline{A}$（A 的补集）：全集中不属于 A 的元素的集合

$A\oplus B$（A 和 B 的对称差）：包含恰属于 A 和 B 之一的那些元素的集合

成员表：显示集合中元素的成员关系的表格

结果 1.5 节表 1-17 给出集合恒等式。

函数（1.6－1.8 节）

术语

从 A 到 B 的函数：一种指派，为 A 中每个元素指派 B 中恰好一个元素

f 的定义域：指集合 A，若 f 是从 A 到 B 的函数

f 的伴域：指集合 B，若 f 是从 A 到 B 的函数

b 是 f 之下 a 的像：$b=f(a)$

a 是 f 之下 b 的原像：$f(a)=b$

f 的值域：f 的像的集合

映上函数，满射：从 A 到 B 的函数，使 B 的每个元素都是 A 中某元素的像

一对一函数，内射：定义域中每个元素的像都不相同的函数

一一对应，双射：既是一对一又是映上的函数

f 的逆：（当 f 是双射时）颠倒由 f 给出的对应关系的函数

$f\circ g$（f 和 g 的组合）：为 x 指派 $f(g(x))$ 的函数

$\lfloor x\rfloor$（底函数）：不超过 x 的最大整数

$\lceil x\rceil$（顶函数）：大于或等于 x 的最小整数

序列：定义域为整数集的子集的函数

串：有限序列

$\sum_{i=1}^{n}a_i$：和 $a_1+a_2+\cdots+a_n$

$\prod_{i=1}^{n}a_i$：乘积 $a_1a_2\cdots a_n$

可数集：有限集合或与正整数集合有一一对应的集合

不可数集：不是可数集的集合

$f(x)$是$O(g(x))$：对某对常数 C 和 k，使得在 $x>k$ 时$|f(x)|\leqslant C|g(x)|$的事实

$f(x)$是$\Omega(g(x))$：对某对常数 C 和 k，使得在 $x>k$ 时$|f(x)|\geqslant C|g(x)|$的事实

$f(x)$是$\Theta(g(x))$：$f(x)$是$O(g(x))$且 $f(x)$ 是$\Omega(g(x))$的事实

结果

实数集合是不可数的。

$\log n!$ 是$O(n\log n)$。

若$f_1(x)$是$O(g_1(x))$，$f_2(x)$是$O(g_2(x))$，那么，$(f_1+f_2)(x)$是$O(g_1(x),g_2(x))$，$(f_1f_2)(x)$是 $O(g_1(x)g_2(x))$。

若 a_0，a_1，…，a_n 是实数，那么 $a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ 是$O(x^n)$和 $\Theta(x^n)$。

复习题

1. a) 定义命题的否定。
   b) “这是一门烦人的课程”的否定是什么？
2. a) (用真值表) 定义命题 p 和 q 的析取、合取、异或、蕴含和双向蕴含。
   b) “今晚我去看电影”和“我将完成离散数学作业”的析取、合取、异或、蕴含和双向蕴含是什么？
3. a) 给出至少五种用汉语表达蕴含 $p\rightarrow q$ 的方式。
   b) 定义蕴含的反蕴含和倒置。
   c) “如果明天阳光明媚，我将到林中散步。”给出此蕴含语句的反蕴含和倒置。
4. a) 两个命题逻辑等价的含义是什么？
   b) 描述证明两个复合命题逻辑等价的不同方法。
   c) 至少用两种方法证明$\neg p\vee(r\rightarrow\neg q)$和$\neg p\vee\neg q\vee\neg r$ 等价。
5. (依赖于 1.2 节的练习)
   a) 给出一个真值表，解释怎样用析取范式构造一个复合命题使其真值表就是你给出的真值表。
   b) 解释为什么 a) 说明运算符$\wedge$，$\vee$和$\neg$是功能完全的。
   c) 是否有一个运算符使得只含这个运算符的集合是功能完全的？
6. 什么是谓词$P(x)$的全称和存在量化？什么是它们的否定？
7. a) 量化语句$\exists x\,\forall y P(x,y)$和$\forall y\,\exists x P(x,y)$的区别何在？其中$P(x,y)$为谓词。
   b) 给出谓词$P(x,y)$的一个实例，使$\exists x\,\forall y P(x,y)$和$\forall y\,\exists x P(x,y)$有不同的真值。
8. a) 定义两个集合的并集、交集、差集和对称差。
   b) 什么是正整数集合和奇整数集合的并集、交集、差集和对称差？
9. a) 定义两个集合相等。
   b) 描述证明两个集合相等的方法。
   c) 至少给出证明$A-(B\cap C)$和$(A-B)\cup(A-C)$相等的两种不同方法。
10. 解释逻辑等价和集合相等之间的关系。
11. a) 定义集合 S 的基数$|S|$。

b）给出计算$|A\cup B|$的一个公式，其中 A 和 B 为集合。

12. a）定义集合 S 的幂集合。

    b）集合 S 的幂集合何时为空集？

    c）含 n 个元素的集合 S 的幂集合有多少个元素？

13. a）定义函数的定义域、伴域和值域。

    b）令 $f(n)=n^2+1$ 为从整数集合到整数集合的函数。它的定义域、伴域和值域是什么？

14. a）给出从正整数集合到正整数集合的函数为一对一函数的含义。

    b）给出从正数集合到正整数集合的函数为映上函数的含义。

    c）给出一个从正整数集合到正整数集合的函数的实例，使它既是一对一的，又是映上的。

    d）给出一个从正整数集合到正整数集合的函数的实例，使它是一对一的，但不是映上的。

    e）给出一个从正整数集合到正整数集合的函数实例，使它不是一对一的，但是映上的。

    f）给出一个从正整数集合到正整数集合的函数实例，使它既不是一对一的，也不是映上的。

15. a）定义函数的反函数。

    b）什么样的函数有反函数？

    c）从整数集合到整数集合的函数 $f(n)=10-n$ 有反函数吗？如果有，其反函数是什么？

16. a）定义从实数集合到整数集合的底函数和顶函数。

    b）对哪些实数 x 有$\lfloor x\rfloor=\lceil x\rceil$？

17. a）用求和符号表示从 2 的幂 2^0 到 2 的幂 2^n 的和。

    b）a）中表示的和的值是什么？

18. a）集合可数的含义是什么？给出一个准确的定义。

    b）负整数集合可数吗？为什么？

    c）分母大于 3 的有理数集合可数吗？为什么可数或不可数？

    d）2 和 3 之间的实数集合可数吗？为什么？

19. a）给出$f(n)$是$O(g(n))$这一事实的定义，其中$f(n)$和 $g(n)$是从正整数集合到实数集合的函数。

    b）用$f(n)$是$O(g(n))$这一事实的定义直接证明或否定 $n^2+18n+107$ 是$O(n^3)$。

    c）用$f(n)$是$O(g(n))$这一事实的定义，直接证明或否定 n^3 是$O(n^2+18n+107)$。

20. a）假定一个函数是几个不同项的和，且每个项都是若干函数的乘积，怎样给这个函数作大 O 估计？

    b）给函数$f(n)=(n!+1)(2^n+1)+(n^{n-2}+8n^{n-3})(n^3+2^n)$一个大 O 估计。在估计$f(x)$是$O(g(x))$中，用阶尽可能小的一个简单函数作$g(x)$。

补充练习

1. 令 p 为命题"我将完成本书中的每一道练习"，q 为命题"这门课程我将得'A'"。将下列各项表示为 p 和 q 的组合。

   a）只有完成本书中的每道练习，这门课程我才能得'A'。

b）这门课程我将得‘A’，而且我将完成本书中每一道练习。

c）或者这门课程我将得‘A’，或者我不完成本书中的每一道练习。

d）这门课程我得‘A’的充分必要条件是我完成本书中每一道练习。

2. 求复合命题$(p \vee q) \rightarrow (p \wedge \neg r)$的真值表。

3. 证明下列命题为永真式。

a) $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$

b) $((p \vee q) \wedge \neg p) \rightarrow q$

4. 给出下列蕴含关系的反蕴含和倒置。

a）如果今天下雨，我就开车上班。

b）如果$|x| = x$，那么$x \geq 0$。

c）若n大于3，那么n^2大于9。

5. 用命题变量p，q，r和s构造一个复合命题，使它在这些命题变量中恰有三个为真时取真值，其他情况下为假。

6. 令$P(x)$为语句“学生x会微积分”，$Q(y)$为“y班上有个学生会微积分”。用$P(x)$和$Q(y)$的量化表示下列各项。

a）某个学生会微积分。

b）不是每个学生都会微积分。

c）每个班上都有一个学生会微积分。

d）每个班上的每个学生都会微积分。

e）至少有一个班没有学生会微积分。

7. 令$P(m, n)$为语句“m除尽n”，其中变量m和n的论域均为正整数集合。给出下列命题的真值。

a) $P(4,5)$　　b) $P(2,4)$

c) $\forall m \forall n P(m, n)$　　d) $\exists m \forall n P(m, n)$

e) $\exists n \forall m P(m, n)$　　f) $\forall n P(1, n)$

8. 令$P(x, y)$为命题函数。证明蕴含关系$\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$为永真式。

9. 令$P(x)$和$Q(x)$为命题函数。求证$\exists x(P(x) \rightarrow Q(x))$和$\forall x P(x) \rightarrow \exists x Q(x)$有同样的真值。

10. 如果$\forall y \exists x P(x, y)$为真，$\exists x \forall y P(x, y)$是否必定也为真？

11. 如果$\forall x \exists y P(x, y)$为真，$\exists x \forall y P(x, y)$是否必定也为真？

12. 找出下列语句的否定。

a）如果今天下雪，那么我明天去滑雪。

b）班上每个人都懂数学归纳法。

c）班上有些学生不喜欢离散数学。

d）每堂数学课都有某个学生上着课就睡着了。

13. 用量词表示“班上每个学生都选修过数学学院每个系的某门课”。

14. 用量词表示“在美国某学院的校园里有座楼的每间屋子都漆成了白色。”

15. 令A为英语中含字母x的所有单词的集合，B是英语中含字母q的所有单词的集合。把下列各集合表示为A和B的组合。

a）英语中不含字母 x 的单词的集合。

b）英语中既含字母 x 又含字母 q 的单词的集合。

c）英语中含字母 x 但不含字母 q 的单词的集合。

d）英语中不含字母 x 或字母 q 的单词的集合。

e）英语中含 x 或含 q 但不同时含 x 和 q 的单词的集合。

16. 求证：若 A 是 B 的子集，则 A 的幂集是 B 的幂集的子集。

17. 假设集合 A 的幂集是集合 B 的幂集的子集。A 是否一定是 B 的子集？

18. 令 $\mathbf{E}$ 表示偶数集合，$\mathbf{O}$ 表示奇数集合，$\mathbf{Z}$ 依旧表示所有整数的集合。求下列各项：

a）$\mathbf{E}\cup\mathbf{O}$　　b）$\mathbf{E}\cap\mathbf{O}$

c）$\mathbf{Z}-\mathbf{E}$　　d）$\mathbf{Z}-\mathbf{O}$

19. 如果 A 是集合，U 是全集，求证

a）$A\cap\overline{A}=\emptyset$　　b）$A\cup\overline{A}=U$

20. 若 A 和 B 为集合，求证

a）$A=A\cap(A\cup B)$　　b）$A=A\cup(A\cap B)$

21. 求证：若 A，B 为集合，则 $A-(A-B)=A\cap B$。

22. 令 A，B 为集合。求证 $A\subseteq B$ 当且仅当 $A\cap B=A$。

23. 令 A，B，C 为集合，求证 $(A-B)-C$ 不一定等于 $A-(B-C)$。

24. 假定 A，B，C 为集合。证明或否定 $(A-B)-C=(A-C)-B$。

25. 假定 A，B，C，D 为集合。证明或否定 $(A-B)-(C-D)=(A-C)-(B-D)$。

26. 证明若 A，B 为有限集合，则 $|A\cap B|\leqslant|A\cup B|$。什么条件下这一关系成为等式？

27. 令 A，B 为有限全集 U 中的集合。按从小到大的顺序排列下面各组数。

a）$|A|$，$|A\cup B|$，$|A\cap B|$，$|U|$，$|\emptyset|$

b）$|A-B|$，$|A\oplus B|$，$|A|+|B|$，$|A\cup B|$，$|\emptyset|$

28. 令 A，B 为有限全集 U 的集合。求证 $|\overline{A}\cap\overline{B}|=|U|-|A|-|B|+|A\cap B|$。

29. 令 f 和 g 分别是从 $\{1, 2, 3, 4\}$ 到 $\{a, b, c, d\}$ 和从 $\{a, b, c, d\}$ 到 $\{1, 2, 3, 4\}$ 的函数，使得 $f(1)=d, f(2)=c, f(3)=a, f(4)=b$ 和 $g(a)=2, g(b)=1, g(c)=3, g(d)=2$。

a）f 是一对一的吗？g 是一对一的吗？

b）f 是映上的吗？g 是映上的吗？

c）f 或 g 是否有反函数？若有，求其反函数。

30. 令 f 为从集合 A 到集合 B 的一对一函数。令 S 和 T 为 A 的子集。求证 $f(S\cap T)=f(s)\cap f(T)$。

31. 给出一个例子说明如果 f 不是一对一的，那么练习 30 中的等式可能不成立。

32. 求证：若 n 为整数，则 $n=\lceil n/2\rceil+\lfloor n/2\rfloor$。

33. 求下面几个量的值。

a）$\sum_{i=0}^{3}(\sum_{j=0}^{4}ij)$　　b）$\prod_{i=1}^{4}(\sum_{j=0}^{3}j)$

c）$\sum_{i=1}^{5}(\sum_{j=0}^{i}1)$　　d）$\prod_{i=1}^{3}(\prod_{j=0}^{i}j)$

34. 0 和 1 之间的无理数集合是可数的吗？给出理由。

** 35. 一个实数称为代数数如果它是某个整系数多项式的根。证明只有可数多个代数数。[提示：利用 n 阶多项式至多只有 n 个不同的根这一事实。]

36. 求证 $8x^3+12x+100\log x$ 是$O(x^3)$。

37. 给$(x^2+x(\log x)^3)\cdot(2^x+x^3)$一个大 O 估计。

38. 求$\sum_{j=1}^{n}j(j+1)$的一个大 O 估计。

*39. 证明 $n!$ 不是$O(2^n)$。

*40. 证明 n^n 不是$O(n!)$。

计算机题目

按给定的输入和输出写程序。

1. 已知命题 p 和 q 的真值，求它们的合取、析取、异或、蕴含和双向蕴含的真值。
2. 已知长度为 n 的两个位串，求它们的按位 AND、按位 OR 及按位 XOR。
3. 已知模糊逻辑中命题 p 和 q 的真值，求 p 和 q 的析取和合取的真值（参看 1.1 节练习 31－33）。
4. 已知含 n 个元素的某集合的子集 A 和 B，用位串求 $\overline{A}$，$A\cup B$，$A\cap B$，$A-B$ 和 $A\oplus B$。
5. 已知同一全集的多重集 A 和 B，求 $A\cup B$，$A\cap B$，$A-B$ 和 $A+B$（参看 1.5 节练习 47 的开场白）。
6. 已知模糊集合 A 和 B，求 $\overline{A}$，$A\cup B$ 和 $A\cap B$（参看 1.5 节练习 49 的开场白）。
7. 已知从 $\{1, 2, \cdots, n\}$ 到整数集合的函数 f，判断 f 是否是一对一的。
8. 已知从 $\{1, 2, \cdots, n\}$ 到它自己的函数 f，判断 f 是否是映上的。
9. 已知从 $\{1, 2, \cdots, n\}$ 到它自己的双射 f，求 f^{-1}。
10. 已知序列 $a_1, a_2, \cdots, a_n$ 的项，求$\sum_{j=1}^{n}a_j$ 和$\prod_{j=1}^{n}a_j$

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 使 $n!$ 是不超过 100 位十进数和 1000 位十进数的 n 的最大值是什么？
2. 前 25 个正整数 n 中每一个 $n!$ 的十进表示有多少个尾部 0？能给出一个表示 $n!$ 的十进表示中尾部 0 个数的公式吗？（参看 2.3 节。）
3. 计算从集合 S 到集合 T 的一对一函数的个数，其中 S 和 T 为各种大小的有限集合。能给出计算这种函数个数的公式吗？（在第 4 章有这样一个公式。）
4. 计算从集合 S 到集合 T 的映上函数的个数，其中 S 和 T 是各种大小的有限集合。能给出计算这种函数个数的公式吗？（第 5 章有这样一个公式。）
5. 我们知道当 b 和 d 为正整数且 $d\geqslant 2$ 时，n^b 是$O(d^n)$。对下列各数值集合，给出常数 C 和 k，使 $x>k$ 时，$n^b\leqslant Cd^n$：

$$b=10,\ d=2;\quad b=20,\ d=30;\quad b=1000,\ d=7$$

写作题目

用课本以外的资料，按下列要求写成短文。

1. 描述模糊逻辑怎样用于实际应用。参考为一般读者写的一本或几本最近出版的模糊逻辑书。
2. 读一些卡罗尔（Lewis Carroll）关于符号逻辑的作品。详细描述他用于表示逻辑论证的一个模型。
3. 讨论如何建立一个公理集合论才能避免罗索悖论。（参看 1.4 节练习 26。）
4. 研究函数概念是从哪里首先提出来的，描述一下这一概念最初是怎样应用的。
5. 解释为什么《整数序列大全》（*The Encyclopedia of Integer Sequences*）［SIP195］对各种各样的人都有用。此外，再说明这本大全中几个不常见的序列是怎样产生的。
6. 描述一下怎样把集合基数的概念推广到无限集合。
7. 查一查超越数的定义。说明怎样证明这种数的存在以及怎样构造这种数。哪些有名的数可以证明是超越数？
8. 查一查巴赫曼最初是怎样引入大 O 符号的。说明他和其他人是怎样使用这一符号的。

第2章 基础：算法、整数和矩阵

许多问题都可以当做一般问题的特例来解决。例如，考虑寻找序列 101，12，144，212，98 中的最大整数的问题。这是寻找整数序列中的最大整数的问题的一个特例。为解决这种一般问题，必须给出一个算法，算法规定解题的一串步骤。本书将介绍解决各种不同类型问题的算法。例如，求两个整数的最大公约数的算法，产生有限集所有排序的算法，检索列表的算法，求网络上两顶点间最短路径的算法，等等。与算法有关的一个重要因素是其计算复杂性，即用这一算法解决一定规模的问题需要什么计算机资源？本章将阐明怎样分析算法的复杂性。

整数集合是离散数学的基础。特别是整数除法的概念是计算机算术的基础。我们将简要回顾一下数论的几个重要概念，以及整数与其性质的研究。将要学习与整数有关的几个重要算法，包括求最大公约数的欧几里德算法，这是3千多年前首次给出的算法。整数可以用任何大于1的正整数为基来表示。贯穿计算机科学的二进制展开即是以2为基的表示。本章将学习以 b 为基的整数表示，并给出求这些表示的算法。同时要讨论用于整数算术的算法，这是称为算法的第一批过程。本章还要介绍几个重要的数论应用。例如，我们将利用数论对信息加密，产生伪随机数，以及为计算机文件分配内存地址。一度被认为是最纯粹的学科数论已成为计算机和网络安全的实质性工具。

矩阵在离散数学中用于表示各种离散结构。我们要回顾矩阵的基本内容以及表示关系和图所需要的矩阵算术。矩阵算术将用于与这些结构有关的大量算法。

2.1 算法

2.1.1 引言

离散数学中有多种一般类型的问题。例如：已知一串整数，求最大的一个；已知一个集合，列出其所有子集；给定一个整数集合，把它们从小到大排成序列；已知一个网络，找两个顶点间的最短路径。遇到这样一个问题时，首先要做的就是构造一个模型，把问题翻释成数学语言。在这种模型中用到的离散结构包括集合、序列和函数——第1章讨论过的结构，以及置换、关系、图、树、网络和有限状态机等其他结构——这些概念将在以后的章节中讨论。

建立合适的数学模型只是解题的第一步。完整的解还需要如何利用这一模型解决一般性问题的方法。理想的情况是，需要一个过程，它能够遵循一串步骤找出所求的解。这一串步骤就称为算法。

定义 1 算法是进行一项计算或解一道题的准确指令的有限集。

算法（algorithm）这一术语是对 al-Khowarizmi（奥尔科瓦里兹米）[○]这一名字的讹用。奥尔科瓦里兹米是 9 世纪时的一位阿拉伯数学家。他关于印地数字的一本书是现代十进制符号的基础。最初 algorism 一词用于表示使用十进制符号做算术运算的规则。18 世纪时 algorism 演变为 algorithm。随着人们对计算机器兴趣的增长，算法的概念有了更广的含义，已经不仅包含算术运算的过程，而且包含所有确定的解题过程。（我们将在 2.4 节讨论整数算术运算的算法。）

本书将讨论解决各种各样问题的算法。这一节要用找有限整数序列中的最大整数这一问题解释算法的概念和算法具有的性质。还要给出在有限集合中找一个特定元素的算法。以后各节将讨论求两个整数的最大公约数的过程，求网络上两点间的最短路径的过程，矩阵相乘的过程，等等。

例 1 描述求有限整数序列中的最大值的算法。

尽管求有限序列最大元素的问题相对而言很平凡，但它能提供对算法概念很好的说明。此外需要求有限整数序列中最大整数的情况各不相同。例如，大学可能要找出几千名学生参加的竞赛的最高分。体育组织可能要确定每月成绩最好的运动员。我们希望开发一个只要出现求有限整数序列最大元素问题时就可以使用的算法。

可以用几种不同的方式给出解这一问题的过程。一种方法是直接用一般语言描述使用的一串步骤。我们现在就给出这样一个解。

例 1 的解：采取下面的步骤。

1）置临时最大值等于序列中第一个整数。（临时最大值将在本过程的每一阶段都等于已检查过的整数中的最大整数。）

2）将序列中下一整数与临时最大值比较，如果它大于临时最大值，置临时最大值为这一整数。

3）如果序列中还有其他整数，重复前一步骤。

4）在序列中没有留下可比的整数时停止，此刻的临时最大值就是序列中的最大整数。

■

一个算法也可以用计算机语言描述。但是，这样做时只能用语言所允许指令。这常常会使算法的描述变得复杂，而且难于理解。再则，有许多通用的程序设计语言，我们不希望从中选用某一个。本书也就不用任何一个特定的计算机语言描述算法，而是使用一种伪码的形式。（本书中所有算法也用一般语言描述。）伪码提供的是在算法的一般语言描述和它的程序语言实现之间的中间一步。算法步骤用模仿程序语言指令的伪指令描述，不过伪指令可以包括有确切定义的一切运算和语句。以算法的伪码描述为起点可以用任何一个计算机语言产生计算机程序。

○ 奥尔科瓦里兹米（Abu Ja'far Mohammed ibn Musa al-Khowarizmi，公元 780—850）天文学家和数学家，是巴格达一个科学家组织"智慧之家"的成员。奥尔科瓦里兹米（al-Khowarizmi）这一名字的含义是"来自科瓦里米（Khowarizm）镇"。该镇现称卡瓦（Khiva），是乌兹别克斯坦的一部分。奥尔科瓦里兹米写过关于数学、天文学和几何学的书，西欧人首先从他的著作中学习代数。代数（algebra）一词源自 al-jabr，这是他的书的标题《*Kitab al-jabr w'al muquabala*》的一部分。这本书曾译为拉丁文并广泛用作课本。他关于印度数字的书描述了使用这些数字作算术运算的过程。欧洲作者使用了他的名字的一个拉丁讹音来表示用印度数字做算术运算，后来演变为 algorithm（算法）一词。

本书使用的伪码大致上以程序语言 Pascal 为基础。不过我们并不遵循 Pascal 或其他程序语言的语法。此外，任何有确切定义的指令均可用于伪码中。附录 2 给出了本书使用的伪码的细节。必要时读者应参看这一附录。

下面是求有限序列最大元素算法的伪码描述。

算法 1 求有限序列的最大元素

```
Procedure max (a1,a2,…,an: 整数)
max := a1
for i := 2 to n
    if max < ai then max := ai
{max 是最大元素}
```

这一算法首先把序列的首项 a_1 赋给变量 max。"**for**"循环用于逐个检查序列的项。如果某一项大于 max 的当前值，就把它赋给 max，成为 max 的新值。

各种算法有若干共有的性质。在描述算法时记住它们是有用的。这些性质是：

- 输入　算法从一个指定的集合得到输入值。
- 输出　对每个输入值集合，算法都要从一个指定的集合中产生输出值。输出值就是问题的解。
- 确定性　算法的步骤必须是准确定义的。
- 正确性　对每一组输入值算法都应产生正确的输出值。
- 有限性　对集合中的任何输入，算法都应在有限（可能很多）步之后产生所求的输出。
- 有效性　算法的每一步必须能够准确地执行，并在有限时间内完成。
- 通用性　算法过程应适用于要求形式的所有问题，而不只是用于一组特定的输入值。

例 2　说明求有限整数序列最大元素的算法 1 具有上面列出的所有性质。

解　算法 1 的输入是一个整数序列。输出是该序列的最大整数。算法的每一步都是准确定义的，因为只出现赋值、有限循环和条件语句。算法步骤是有限的，因为在序列中的所有整数都被检查过以后它就终止。算法在有限时间内完成，因为每一步要么是比较，要么是赋值。最后算法 1 是通用的，因为它可以用于求任何有限整数序列的最大元素。 ■

2.1.2 搜索算法

在有序表中常会出现为元素定位的问题。例如检查单词拼写的程序要在字典中搜索，字典其实就是单词的有序表。这一类问题称为*搜索问题*。本节将讨论几个搜索算法。在 2.2 节将考查这些算法中各自使用的步数。

一般的搜索问题可以描述如下：在不同元素 a_1，a_2，…，a_n 的表中为元素 x 定位，或判定它不在该表中。这一搜索问题的解是表中等于 x 的项的位置（即若 $x = a_i$，那么 i 就是解），或当 x 不在表中时解为 0。

我们将介绍的第一个算法称为*线性搜索算法*或*顺序搜索算法*。线性搜索算法从比较 x 和 a_1 开始，若 $x = a_1$，那么解就是 a_1 的位置，也就是 1。当 $x \neq a_1$ 时，比较 x 和 a_2。若 $x =$

a_2，解就是 a_2 的位置，也就是 2。当 $x \neq a_2$ 时，比较 x 与 a_3。继续这一过程，逐一比较 x 和表中每一项，直到出现相等，解就是该项的位置，除非不存在相等。如果搜索整个表后都不能为 x 定位，那么解是 0。这一线性搜索算法的伪码如算法 2 所示。

算法 2　线性搜索算法

```
Procedure linear search (x: 整数, a1, a2, …, an: 不同整数)
i := 1
while (i ≤ n 和 x ≠ ai)
    i := i + 1
if i ≤ n then location := i
else location := 0
{location 是等于 x 的项的下标，或是找不到 x 时为 0}
```

现在我们考虑另一个搜索算法。当表中各项以增序出现时可以用这一算法。（例如：若各项为数，它们按从最小到最大排列；如果它们是单词，可以按字典顺序或字母顺序排列。）这第二个算法称为对分搜索算法。它是把要搜索的元素与表的中间项比较。然后这表就分成两个长度相等的较小子表，或两个子表中有一个比另一个少一个项。根据与中间项的比较结果，搜索将限于在两个子表的一个中进行。在下一节我们将证明对分搜索比线性搜索的效率要高得多。下面的例子说明如何进行对分搜索。

例 3　要在表

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22

中搜索 19。第一步把有 16 个项的这个表分成各含 8 个项的两个较小的表，即

1 2 3 5 6 7 8 10　12 13 15 16 18 19 20 22

然后比较 19 和头一个表的最大项。因为 10<19，对 19 的搜索，可以限于包含原表第 9 项到第 16 项的表中。下一步把含 8 个项的这个表分成两个含 4 个项的小表，即

12 13 15 16　　18 19 20 22

因为 16<19（将 19 与第一个表的最大项比较），搜索限于这两个表的第二个，它包含原表的第 13 项到第 16 项。表 18 19 20 22 被分成两个表，即

18 19　　20 22

因为 19 不大于两个表中第一个的最大项，此最大项也是 19，搜索限于第一个表：18 19，这个表包含原表的第 13 和 14 项。下一步，这个两项的表分成各含 1 项的两个表 18 和 19。因为 18<19，搜索限于第二个表：这个表只含原表第 14 项，即 19。现在搜索已经收缩到一项上。与这一项比较，19 定位为原表的第 14 项。■

现在我们给出对分搜索算法的步骤。要在表 $a_1, a_2, \cdots, a_n$ 中搜索整数 x，其中 $a_1 < a_2 < \cdots < a_n$，我们从比较 x 和序列的中间项 a_m 开始，其中 $m = \lfloor (n+1)/2 \rfloor$。（回忆一下，$\lfloor x \rfloor$是不超过 x 的最大整数。）如果 $x > a_m$，搜索可以限制在序列的下半段，即 a_{m+1}，

a_{m+2}，…，a_n。如果 x 不大于 a_m，搜索可限制在序列的上半段，即 a_1，a_2，…，a_m。

现在搜索的范围限于一个不超过$\lceil n/2 \rceil$个元素的表。用同样的过程，比较 x 和这个短表的中间项，然后把搜索限于短表的前半段或后半段。这样重复直到得到只含一个项的表。然后判断这个项是否就是 x。对分搜索算法的伪码由算法 3 给出。

算法 3 对分搜索算法

```
Procedure binary search (x: 整数, a1, a2, …, an: 递增整数)
i := 1 {i 是搜索区间的左端点}
j := n {j 是搜索区间的右端点}
while i < j
begin
    m := ⌊(i + j)/2⌋
    if x > am then i := m + 1
    else j := m
end
if x = ai then location := i
else location := 0
{location 是等于 x 的项的下标，或在找不到 x 时为 0}
```

算法 3 一次次收缩被搜索的序列。在任何阶段都只有从 a_i 开始到 a_j 结束的这些项需要考虑。换言之，i 和 j 分别是留下需检查的最小和最大下标。算法 3 不断收缩需搜索的序列，直到剩下只有一项的序列。在执行这一步时，比较这一项是否等于 x。

练习

1. 如果要在表 1，8，12，9，11，2，14，5，10，4 中找最大值，列出算法 1 使用的所有步骤。
2. 判断下列过程各有算法的什么特性，各缺什么特性。

a）
```
Procedure double (n: 正整数)
while n > 0
    n := 2n
```

b）
```
Procedure divide (n: 正整数)
while n ≥ 0
begin
    m := 1/n
    n := n - 1
end
```

c）
```
Procedure sum (n: 正整数)
sum := 0
while i < 10
```

$sum := sum + i$

d) **Procedure** *choose* (a, b: 整数)

$x := a$ 或 b

3. 设计一个求表中所有整数之和的算法。
4. 设计计算 x^n 的算法，其中 x 是个实数，n 是个整数。[提示：首先给出一个 n 为非负整数时从 1 开始不断乘以 x 来计算 x^n 的过程。然后扩充这一过程，当 n 为负数时，利用 $x^{-n}=1/x^n$ 的事实。]
5. 描述一个交换变量 x 和 y 的值的算法，只许使用赋值。至少需要多少个赋值语句才能完成交换？
6. 描述一个只使用赋值语句的用三元组 (y, z, x) 代替 (x, y, z) 的算法。为此最少需要多少个赋值语句？
7. 列出在序列 1，3，4，5，6，8，9，11 中搜索 9 的所有步骤，使用下述算法：
   a) 线性搜索。
   b) 对分搜索。
8. 列出在练习 7 给出的序列中搜索 7 使用的所有步骤。
9. 给出一个算法，把整数 x 插入按增序排列的整数表 a_1，a_2，…，a_n 中的合适位置。
10. 给出一个求有限自然数序列中最小整数的算法。
11. 给出确定有限整数表中最大元素首次出现位置的算法，表中的整数不必互不相同。
12. 给出确定有限整数表列中最小元素最后出现位置的算法，表列中的整数不必互不相同。
13. 找一个算法计算含有三个整数的集合的最大值、中间值、平均值和最小值。(整数集合的中间值是把这些整数按增序排列时中间元素的值。整数集合的平均值是这些整数之和除以整数个数。)
14. 给出一个求有限整数序列中最大和最小整数的算法。
15. 给出一个算法把任意长整数序列的头三项按增序排列。
16. 给出一个算法求英文句子中最长的单词。(单词指字母串，句子指用空格分隔的一串单词。)
17. 给出一个判断从一个有限集合到另一个有限集合的函数是否为映上的算法。
18. 给出一个判断从一个有限集合到另一个有限集合的函数是否为一对一的算法。
19. 给出一个算法，逐一检查位串中每个字位是否为 1，计算其中 1 字位的个数。
20. 改动算法 3，使对分搜索过程在每一阶段都比较 x 和 a_m，并在 $x=a_m$ 时终止。这一改动后的对分搜索算法有何优越之处？
21. 三分搜索算法为元素在增序整数表中定位，定位方法是依次把表分成长度相等（或尽可能相等）的三个子表，每次都把搜索限制在一个合适的子表中。描述这一算法的步骤。
22. 描述一个为元素在增序整数表中定位的搜索算法，搜索方式是每次把表分成相等（或尽可能相等）的 4 个子表，并把搜索限制在一个合适的子表中。
23. 整数表的一个众数是表中出现次数不比任何别的元素出现次数少的元素。设计一个算法，求非递减整数表的一个众数。
24. 设计一个算法，求非递减整数表的所有众数（练习 23 中定义）。
25. 设计一个算法，求整数序列中第一个与序列中排在它前面的某项相等的项。

26. 设计一个算法，找出有限整数序列中所有那些大于它前面各项之和的项。
27. 设计一个算法，求正整数序列中第一个小于其前项的项。

2.2 算法的复杂性

2.2.1 引言

什么时候算法对问题提供令人满意的解？首先，它必须给出正确的答案。第 3 章将讨论如何说明它正确。其次，它必须有效率，本节讨论算法的效率。

怎样分析算法的效率呢？一种效率度量是计算机按此算法解题所花的时间，另一种度量是计算机实现这一算法需要多大内存，当然都假定输入值的规模是一定的。

这些问题都涉及算法的**计算复杂性**。分析解决特定规模的问题需要的时间是算法的**时间复杂性**。分析需要的计算机内存是算法的**空间复杂性**。在实现一个算法时，时间和空间复杂性的考虑都是实质性的。显然，了解算法是一微秒、一分钟还是 10 亿年才能给出答案是很重要的。类似地，必须能提供所需的内存才能解题，所以必须考虑空间复杂性。

空间复杂性的考虑与实现算法时使用的特定数据结构紧密相关。由于本书对数据结构不做详细讨论，我们不考虑空间复杂性，而将注意力集中在时间复杂性上。

算法的时间复杂性可以在输入规模一定的情况下用算法使用的运算次数来表示。度量时间复杂性使用的运算包括整数比较、整数加法、整数乘法、整数除法或任何其他基本运算。

时间复杂性用运算次数而不是计算机实际使用时间来表示，这是因为在执行基本运算时不同的计算机需要的时间不同。此外，把所有运算分解成计算机使用的基本字位运算是相当复杂的。再者，现存最快的计算机执行字位运算（例如两个字位的加法、乘法、比较或交换）的时间是 10^{-9}秒（1 纳秒），但个人电脑可能需要 10^{-6}秒（1 微秒），作同样的运算时间相差 1000 倍。

我们考虑 2.1 节求有限整数集合最大值的算法 1，用以说明怎样分析算法的时间复杂性。

例 1 描述 2.1 节求集合最大元素的算法 1 的时间复杂性。

解 由于比较是该算法使用的基本运算，就以比较的次数作为其时间复杂性的度量。

要求出以任意顺序列出的 n 个元素集合的最大元素，首先让临时最大元素等于列表中的初始项。然后在做一次比较后判断尚未到达列表的终点。于是临时最大元与第二项比较，如果第二项大，就用第二项的值更新临时最大元。这一过继续下去，对表中的每一项都要加上两次比较，一次判断尚未达到列表终点，另一次判断是否需要更临时最大元。由于对于从第二项到第 n 项的每一项都要用两次比较，再加上一次在 $i=n+1$ 时退出循环的比较，所以应用这一算法时使用的比较次数恰为$2(n-1)+1=2n-1$。因此，求 n 元集合最大值的算法的时间复杂性是$O(n)$，这是用算法使用的比较次数来度量的。 ■

下面将分析搜索算法的时间复杂性。

例 2 描述线性搜索的时间复杂性。

解 该算法使用的比较次数将用来度量时间复杂性。算法中每次循环都做两次比较，一次判断是否已到表的终点，一次比较 x 和表的一个项。最后还要在循环外做一次比较。于

是当 $x = a_i$ 时，最多需要做 $2i+1$ 次比较。当 x 不在表中时，最多需要 $2n+2$ 次比较。在这种情况下，$2n$ 次比较用于判断 x 不等于 a_i，$i=1, 2, \cdots, n$，再加上一次用于脱离循环的比较，和一次循环外的比较。所以当 x 不在表中时，共用 $2n+2$ 次比较。从而线性搜索最多需要 $O(n)$ 次比较。

例 2 中做的这类复杂性分析是*最坏情况分析*。所谓算法的最坏情况，指的是把算法应用于一定规模的问题时最多需要多少次运算。最坏情况分析告诉我们算法需要多少次运算就保证给出问题的解答。

例 3　描述对分搜索算法的时间复杂性。

解　为简化描述，假定表 $a_1, a_2, \cdots, a_n$ 中有 $n = 2^k$ 个元素，其中 k 是非负整数。注意 $k = \log n$。（如果表中元素个数 n 不是 2 的幂，那么这个表可以看作一个有 2^{k+1} 个元素的大表的一部分，其中 $2^k < n < 2^{k+1}$）。因此 2^{k+1} 是大于 n 的 2 的最小幂。）■

在算法的每一阶段，都要比较 i 和 j 来判断待搜索的表是否包含 1 个以上的元素，其中 i 和 j 分别是当前待搜索表的第一项和最后项的位置。如果 $i<j$，就是做一次比较判断 x 是否大于待搜索表的中间元素。

在第一阶段将搜索限于含 2^{k-1} 个元素的表。至此已使用了两次比较。这一过程继续下去，每一阶段都用两次比较把搜索限制在长度减半的表中。换言之，第一阶段当表中含 2^k 个元素时，用两次比较把表减半，在表减为 2^{k-1} 个元素时再用两次比较，当表减为 2^{k-2} 时又使用两次比较，如此等等，直到两次比较后使表长为 $2^1 = 2$。最后当表中只剩一个元素时，再用一次比较确定没有其他元素留在表中，还要一次比较判断这一项是否为 x。

因此，当表中有 2^k 个元素时，作对分搜索需要 $2k+2 = 2\log n + 2$ 次比较。（如果 n 不是 2 的幂，最初的表扩展为含 2^{k+1} 项的表，其中 $k = \lfloor \log n \rfloor$，搜索需要最多 $2\lceil \log n \rceil + 2$ 次比较。）于是对分算法需要最多 $O(\log n)$ 次比较，由此分析可知，在最坏的情况下，对分搜索算法比线性搜索效率高。

除最坏情况分析以外，还有称为*平均情况分析*的另一类重要的复杂性分析。这一类分析是就一定规模的问题对所有输入求解问题使用的平均运算次数。平均情况时间复杂性分析一般比最坏情况分析复杂得多。不过线性搜索算法的平均情况分析不难完成，如例 4 所示。

例 4　描述线性搜索算法的平均情况性能，假定元素 x 在表中。

解　当已知 x 的确在表中时，有 n 类可能的输入。如果 x 是表的第一项，需要比较 3 次，一次判断是否已到表终点，一次比较 x 和第一项，再在循环外比较一次。如果 x 是表的第二项，还要增加 2 次比较，所以总共要比较 5 次。一般来说，若 x 是表的第 i 项，i 次循环中的每一次都要做 2 次比较，外加循环外一次，所以共需要做 $2i+1$ 次比较。于是平均使用比较的次数是

$$\frac{3+5+7+\cdots+(2n+1)}{n} = \frac{2(1+2+3+\cdots+n)+n}{n}$$

在 3.2 节将证明

$$1+2+3+\cdots+n = \frac{n(n+1)}{2}$$

所以线性搜索使用的平均比较次数（在已知 x 的确在表中时）是

$$\frac{2(n(n+1)/2)+n}{n}=n+2$$

这是$O(n)$。

注意 分析中假定了x在被搜索的表中，而且它在一切位置的机会都相等。x可能不在表中，这时也可以对这一算法做平均情况分析（参看本节末的练习 13)。■

表 2-1 中给出了描述算法时间复杂性的几个常用术语。例如，如果算法的时间复杂性用某种指定的运算来度量是$O(b^n)$，$b>1$，就说它有指数复杂性。类似地，时间复杂性为$O(n^b)$的算法有多项式复杂性。线性搜索算法（最坏情况或平均情况）有线性复杂性，而对分搜索算法有对数（最坏情况）复杂性。以上搜索算法都是以使用的比较次数来度量时间复杂性。

能用具有多项式最坏情况复杂性的算法解决的问题称为易处理的，因为只要问题的输入规模合理，我们就可期望算法在相对短的时间内给出解答。不过，如果在大O估计中的多项式次数高（如 100 次），或多项式的系数特别的大，算法可能会花特别长的时间来解题。所以，能用具有多项式最坏情况复杂性的算法来解的问题，即使对于相对较少的输入值，也不能保证能在合理时间内得解答。幸运的是，实践中这种估计用到的多项式的次数和系数都不大。

表 2-1 算法复杂性常用术语

复杂性	术 语	复杂性	术 语
$O(1)$	常数复杂性	$O(n^b)$	多项式复杂性
$O(\log n)$	对数复杂性	$O(b^n)$，$b>1$	指数复杂性
$O(n)$	线性复杂性	$O(n!)$	阶乘复杂性
$O(n\log n)$	$n\log n$ 复杂性		

对不能用最坏情况多项式时间复杂性的算法求解的问题，情况要糟得多。这种问题称为不易处理的。一般来说，需要特别大量的时间来解最坏情况的问题，即使输入值很少也是这样；当然也并非总是如此。不过也有这样的时候，以最坏情况时间复杂性的算法解决实际问题时，大多数情况都比最坏情况快得多。如果我们愿意允许少量情况下问题不能在合理的时间内得到解答，那么平均情况时间复杂性是对算法解题时间长短的更好度量。工业上许多重要的问题都认为是不易处理的，但实践中，对日常生活中出现的所有输入集合都得到问题的解。另一种处理实际应用中出现的不易处理的问题的方法，是不求问题的精确解，而以近似解代替。也许存在求近似解的快速算法，甚至还能保证这些近似解与精确解相差不太大。

甚至有一些可以证明没有解题算法存在的问题。这种问题称为不可解的（相对于可以用某个算法求解的可解问题而言)。第一个证明存在不可解问题的是伟大的英国数学家和计算机科学家图灵。他证明不可解的是停机问题。这个问题的输入是一个程序，加上对这一程序的输入。问题要问的是当这个程序以给定的输入执行时会不会停机。我们将在 3.1 节学习停机问题。(第 10 章有图灵小传及对他某些其他工作的介绍。)

算法复杂性的研究远超出我们在这里能介绍的。要注意，人们相信许多可解的问题都没有多项式最坏情况时间复杂性算法能解答它们，但是一旦有了一个解答，却可以以多项式时间来验证。能以多项式时间验证解的问题称为属于 NP 类（易处理的问题属于 P 类）。还有

称为 NP-完全问题的一类重要问题。这类问题具有这样的性质，只要其中任何一个问题都能用一个多项式时间最坏情况算法来解，那么所有这些问题都能用多项式时间最坏情况算法解答。尽管人们做了广泛的研究，并没有找到多项式时间最坏情况的算法能解这一类问题中的任何一个。虽然没有证明，但人们普遍接受的是，没有 NP 完全问题能用多项式时间解。要更多了解算法复杂性的信息，参阅书后为本节列出的文献，包括［CoLeRi 90］。

注意，算法时间复杂性的大 O 估计表达的是解题需要的时间如何随输入数据规模的增长而改变。实践中使用的是能得到证明的最好估计（即用最小函数参照的估计。）不过时间复杂性的大 O 估计不能直接翻译成计算机使用的实际时间量。一个原因是，大 O 估计 $f(n)$ 是 $O(g(n))$ 表示的是有常数 C 和 k，使得在 $n>k$ 时 $f(n)\leqslant Cg(n)$，其中 $f(n)$ 是算法的时间复杂性，$g(n)$ 是参照函数。所以不知道不等式中的常数 C 和 k，就不能用这一估计来判定算法使用的运算次数的上界。此外前文已经说过，每次运算所需要的时间还因运算类型不同和使用的计算机不同而有差异。（还要注意，算法时间复杂性的大 O 估计只能对算法需要的最坏情况时间以输入值规模的函数的形式提供上界，而不提供下界。要提供下界，就应使用大 Θ 函数估计。不过为简单起见，在讨论算法时间复杂性时我们将用大 O 估计，同时懂得大 Θ 估计能提供更多的信息。）

不过，如果算法使用的所有运算都能归结为计算机使用的字位运算，那么算法解答一定规模的问题所需的时间是能够确定的。表 2-2 给出的是算法解答各种规模的问题需要的时间，算法使用的字位运算次数如表中所示。需要的时间超过 10^{100} 年时，表中用星号表示。（在 2.4 节将讨论整数加法和乘法使用的字位运算次数。）在构造这一表格时，假定了每次字位运算需要的时间是 10^{-9} 秒，这是今天最快的计算机需要的时间。将来有了更快的计算机，这些时间将会减少。

表 2-2　算法使用的计算机时间

问题规模	使用的字位运算					
n	$\log n$	n	$n\log n$	n^2	2^n	$n!$
10	$3\times10^{-9}s$	$10^{-8}s$	$3\times10^{-8}s$	$10^{-7}s$	$10^{-6}s$	$3\times10^{-3}s$
10^2	$7\times10^{-9}s$	$10^{-7}s$	$7\times10^{-7}s$	$10^{-5}s$	$4\times10^{13}yr$	*
10^3	$1.0\times10^{-8}s$	$10^{-6}s$	$1\times10^{-5}s$	$10^{-3}s$	*	*
10^4	$1.3\times10^{-8}s$	$10^{-5}s$	$1\times10^{-4}s$	$10^{-1}s$	*	*
10^5	$1.7\times10^{-8}s$	$10^{-4}s$	$2\times10^{-3}s$	$10s$	*	*
10^6	$2\times10^{-8}s$	$10^{-3}s$	$2\times10^{-2}s$	17min	*	*

重要的是了解计算机求解一个问题需要多长时间。例如，如果算法需要 10 小时，也许值得花费这些机时（和金钱）来解决问题。但如果算法解一道题需要百亿年，就没有理由消耗资源来实现这一算法。现代技术最有趣的现象之一是计算机速度和内存空间的巨大增长。减少计算机解题时间的另一重要因素是并行处理，这是同时执行几个运算序列的技术。由于计算速度的增加，计算机内存的增加，再加上使用能做并行处理的算法，5 年前认为无法解的问题现在可以当做日常事务处理了，而且可以肯定这句话在 5 年以后仍然成立。

练习

1. 用 2.1 节练习 10 给出的算法求含 n 个自然数的序列中的最小自然数，要使用多少次比

较？

2. 写一个算法，把任意长的表中前 4 项按增序排列。证明算法使用的比较次数的时间复杂性是O(1)。

3. 假定已知一个元素是含 32 个元素的一个表的前 4 个元素之一。用线性搜索还是对分搜索能更快地为此元素定位？

4. 给出计算 x^{2^k} 使用的乘法次数，假定从 x 开始逐次做平方（求 x^2，x^4，等等）。这样计算 x^{2^k} 是否比连乘 x 相应次数的效率高？

5. 给出下述算法所使用的比较次数的大 O 估计：检查位串的每一位是否为 1，计算位串中 1 的个数（参看 2.1 节练习 19）。

*6. a）证明下面的算法给出的是位串 S 中 1 的个数。

```
Procedure bitcount (S:位串)
count := 0
while S≠0
begin
    count := count + 1
    S := S ∧ (S - 1)
end {count 是 S 中 1 的个数}
```

其中 $S-1$ 是把 S 中最右边的 1 改为 0，同时把这一位右边的所有 0 均改为 1 得到的位串。（回忆一下，$S\wedge(S-1)$是 S 和 $S-1$ 的按位 *AND*。）

b）上述算法要做多少次按位 *AND* 运算才能求出位串 S 中 1 的个数？

7. 计算多项式$a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ 在 $x=c$ 的值的传统算法可以用伪码表示为：

```
Procedure polynomial (c, a0, a1, …, an:实数)
power := 1
y := a0
for i := 1 to n
begin
    power := power * c
    y := y + ai * power
end {y = an c^n + a(n-1) c^(n-1) + … + a1 c + a0}
```

其中 y 的最终值即是该多项式在 $x=c$ 的值。

a）按上述算法一步步计算 $3x^2+x+1$ 在 $x=2$ 的值。

b）准确给出计算 n 阶多项式在 $x=c$ 的值使用的乘法和加法的次数。（不要计算增加循环变量的值所做的加法。）

8. 有一个算法计算多项式的值比上道练习题中给出的传统算法效率高（以使用的乘法和加法次数度量），这个算法称为 Horner 法。下面的伪码说明怎样用这一方法计算 $a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ 在 $x=c$ 的值。

```
Procedure Horner (c, a0, a1, …, an : 实数)
y := an
for i := 1 to n
   y := y * c + a(n-i)
{y = an c^n + a(n-1) c^(n-1) + … + a1 c + a0}
```

a) 按上述算法一步步计算 $3x^2+x+1$ 在 $x=2$ 的值。

b) 准确给出此算法计算 n 阶多项式在 $x=c$ 的值使用的乘法和加法的次数。(不要计算增加循环变量的值所做的加法。)

9. 假定每次字位运算能用 10^{-9} 秒完成，需使用 $f(n)$ 次字位运算的算法在 1 秒钟内能解决多大的问题？设 $f(n)$ 的值如下：

   a) $\log n$　　b) n　　c) $n\log n$　　d) n^2　　e) 2^n　　f) $n!$

10. 假定算法解答规模为 n 的问题需 $2n^2+2^n$ 次字位运算，每次字位运算需 10^{-9} 秒，对于下面给出的 n 的值，计算算法解题花的时间。

    a) 10　b) 20　c) 50　d) 100

11. 假定每次字位运算需要的时间如下，使用 2^{50} 字位运算的算法需要多少时间？

    a) 10^{-6} 秒　　b) 10^{-9} 秒　　c) 10^{-12} 秒

12. 判定下述情况需要的最少比较次数，也就是最好情况性能。

    a) 用 2.1 节的算法 1 求 n 个整数的序列的最大值。

    b) 用线性检索在 n 个元素的表中为一个元素定位。

    c) 用对分搜索在 n 个元素的表中为一个元素定位。

13. 分析线性搜索的平均情况性能，假定恰有一半的情况 x 不在表中，而且当 x 在表中时，它出现在表的任何位置的可能性能一样。

14. 对一个算法和它涉及的某种运算来说，如果不存在能用更少的此种运算解同一问题的其他算法，就说该算法对此种运算是最优的。

    a) 证明 2.1 节算法 1 对于整数比较的次数而言是最优的。(注意：不考虑用于循环控制的比较。)

    b) 对于整数比较次数而言，线性搜索是最优的吗？(不计循环控制用的比较。)

15. 描述一下 2.1 节练习 21 给出的三分搜索算法用比较次数度量的最坏情况时间复杂性。

16. 描述一下 2.1 节练习 22 中给出的搜索算法用比较次数度量的最坏情况时间复杂性。

17. 分析你为 2.1 节练习 23 设计的算法的最坏情况时间复杂性，该算法为非递减整数表的一个众数定位。

18. 分析你为 2.1 节练习 24 设计的算法的最坏情况时间复杂性，该算法为非递减整数表所有众数定位。

19. 分析你为 2.1 节练习 25 设计的算法的最坏情况时间复杂性，该算法求整数序列中第一个与它前面某项相等的项。

20. 分析你为 2.1 节练习 26 设计的算法的最坏情况时间复杂性，该算法求序列中所有那些大于其前面各项之和的项。

21. 分析你为 2.1 节练习 27 设计的算法的最坏情况时间复杂性，该算法求序列中第一个小

于前一项的项。

2.3　整数和除法

2.3.1　引言

离散数学中涉及整数及其性质的这一部分属于数学中称为*数论*的分支。本节是介绍数论的三节中的第一节，先要复习一下数论中的某些基本概念，包括可除性，最大公约数以及模运算。在2.4节将介绍数论中的几个重要的算法，并把2.1节和2.2节中关于算法及其复杂性的内容与本节引入的概念结合在一起。例如，我们将引入求两个正整数的最大公约数的算法及用二进制展式做计算机算术运算的算法。最后在2.5节继续讨论数论，介绍一些重要的结果以及这些结果在计算机算术和密码学研究信息保密的应用。

本节将要展开的思想以可除性为基础。从可除性建立的一个重要概念是素数。素数是大于1且只能被1和它自己整除的整数。判断一个整数是否为素数对密码学应用来说是很重要的。数论中的一个重要的定理"算术基本定理"断言，每个正整数都能唯一地分解为素数的乘积。把整数分解为素数因子在密码学中是很重要的。用一个正整数除一个整数得到商和余数。贯穿计算机科学的模运算处理的就是这些余数。本节将讨论模运算的三个应用：生成伪随机数，为文件分配内存地址，以及为信息加密和解密。

2.3.2　除法

当一个整数被另一个整数除的时候，商可能是也可能不是整数。例如，$12/3=4$ 是整数，而 $11/4=2.75$ 不是整数。这样就导至下面的定义。

定义1　如果 a 和 b 是整数，$a\neq 0$，若有整数 c 使 $b=ac$，就说 a 整除 b。在 a 整除 b 时，a 是 b 的一个因子，b 是 a 的倍数。符号 $a\mid b$ 表示 a 整除 b。当 a 不能整除 b 时写成 $a\nmid b$。

图2-1中的数轴显示的是那些能被正整数 d 整除的整数。

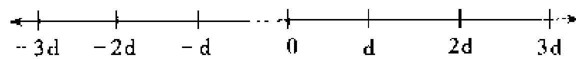


图2-1　可被正整数 d 整除的整数

例1　判断是否有 $3\mid 7$ 和 $3\mid 12$。

解　由于 $7/3$ 不是整数，所以 $3\nmid 7$。另一方面因为 $12/3=4$，于是 $3\mid 12$。 ■

例2　令 n 和 d 为正整数。不超过 n 的正整数中有多少个能被 d 整除？

解　能被 d 整除的正整数都是 dk 形的，其中 k 是正整数。因此不超过 n 的正整数中能被 d 整除的整数的个数，等于能使 $0<dk\leqslant n$ 或 $0<k\leqslant n/d$ 的整数 k 的个数。因此，有 $\lfloor n/d\rfloor$ 个正整数，既不超过 n，又能被 d 整除。 ■

定理1给出了整数可除性的某些基本性质。

定理1　令 a，b，c 为整数。有如下结论：

1）若 $a\mid b$ 和 $a\mid c$，则 $a\mid(b+c)$；

2）若 $a\mid b$，那么对所有整数 c 都有 $a\mid bc$；

3) 若 $a \mid b$，$b \mid c$，则 $a \mid c$。

证　假定 $a \mid b$ 和 $a \mid c$。从可除性定义知有整数 s 和 t，使 $b = as$ 和 $c = at$。因此

$$b + c = as + at = a(s + t)$$

于是 a 整除 $b + c$。这就证明了定理中的 1)。结论 2)和 3)的证明留给读者作练习。 □

2.3.3 素数

大于 1 的每个正整数都至少能被两个整数整除，因为正整数可以被 1 和它自已整除。恰有两个不同的正整数因子的整数称为*素数*。

定义 2　大于 1 的正整数 p 称为素数，如果 p 仅有的正因子是 1 和 p。大于 1 又不是素数的正整数称为*合数*。

例 3　由于 1 和 7 是 7 仅有的正因子，7 是素数。9 是合数，因为它能被 3 整除。 ■

小于 100 的素数是 2，3，5，7，11，13，17，19，23，29，31，37，41，43，47，53，59，61，67，71，73，79，83，89 和 97。

正如算术基本定理所示，素数是构造正整数的积木。定理的证明将在 3.2 节给出。

定理 2　*算术基本定理*　每个正整数都可以唯一地表示为素数的乘积，其中素数因子从小到大依次出现。(这里说的乘积可以有 0 个、1 个或多个素因子。)

下一个例子给出了几个整数的素因子分解。

例 4　100，641，999 和 1024 的素因子分解如下：

$$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 5^2$$

$$641 = 641$$

$$999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$$

$$1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$$ ■

往往很重要的是证明一个给定的整数是素数。例如在密码学中为信息加密的某些方法用到大素数。根据下面的定理可以得到证明整数为素数的一个过程。

定理 3　如果 n 是合数，那么 n 必有一个小于或等于 $\sqrt{n}$ 的素因子。

证　如果 n 是合数，它有一个因子 a，使 $1 < a < n$，于是 $n = ab$，其中 a 和 b 都是大于 1 的正整数。这样就有 $a \leqslant \sqrt{n}$ 或 $b \leqslant \sqrt{n}$，否则 $ab > \sqrt{n} \cdot \sqrt{n} = n$。所以 n 有一个不大于 $\sqrt{n}$ 的正因子。这个因子或是素数，或根据算术基本定理有素因子。无论哪种情况，n 都有小于或等于 $\sqrt{n}$ 的素因子。 □

从定理 3 知，如果整数不能被小于或等于其平方根的素数整除，它就是素数。下面的例子就用这一结论证明 101 是素数。

例 5　证明 101 是素数。

解　不超过 $\sqrt{101}$ 的仅有的素数是 2，3，5 和 7。因为 101 不能被 2，3，5 和 7 整除

(101 被这些数除的商都不是整数)，所以 101 是素数。 ■

由于每个整数都有素因子分解。若能有求解这一素因子分解的算法，一定会很有用的。考虑求解整数 n 的素因子分解的问题。从最小素数 2 开始，从小到大用一个个素数去除 n。如果 n 有素因子，那么按定理 3 必能找到一个不超过 $\sqrt{n}$ 的素因子 p。所以，如果找不到不超过 $\sqrt{n}$ 的素因子，n 为素数。否则，如果找到素因子 p，可以对 n/p 继续这样做。注意 n/p 没有小于 p 的素因子。同样的道理，如果 n/p 没有大于等于 p 且不超过它的平方根的素数因子，它必为素数。否则，如果它有个素因子 q，可以继续对 $n/(pq)$ 做同样的工作。这一过程继续下去直到分解减少到一个素数。下面的例子解释了这一过程。

例 6 求 7007 的素因子分解。

解 求 7007 的素因子分解，从 2 开始用一个个素数除 7007。2，3 和 5 除不尽 7007。但 7 除尽 7007，$7007/7=1001$。下一步从 7 开始用一个个素数除 1001。我们立刻发现 7 还能除尽 1001，$1001/7=143$。从 7 开始，再用一个个素数除 143。虽然 7 不能整除 143，但 11 可以，$143/11=13$。由于 13 为素数。这一过程即告完成。由此得 7007 的素因子分解是 $7\cdot7\cdot11\cdot13=7^2\cdot11\cdot13$。 ■

古代因哲学的原因而研究素数。今天研究素数有着很实际的理由。特别是在密码学中，大素数起着关键性的作用。在 2.5 节我们将会具体讨论这种作用。3.1 节将要证明人们长期以来已经知道的一个事实，即素数的个数是无限的。人们一直希望发现越来越大的素数；差不多近 300 年来已知最大的素数都是形式为 2^p-1 的这种特殊整数，其中 p 也是素数。这种素数称为莫孙尼 (Mersenne)[○] 素数，这是以法国修道士莫孙尼的名字命名的。莫孙尼在十七世纪对这些素数进行了研究。之所以最大已知素数通常都是莫孙尼素数，是因为有个特别有效的称为卢卡斯—来赫莫 (Lucas-Lehmer) 测试的方法可以判断 2^p-1 是否为素数。而且当前还不可能以差不多同样的速度判断一个不是某种特殊形式的整数是否为素数。

例 7 整数 $2^2-1=3$，$2^3-1=7$ 和 $2^5-1=31$ 都是莫孙尼素数，而 $2^{11}-1=2047$ 不是莫孙尼素数，因为 $2047=23.89$。 ■

自从发明了计算机，发现莫孙尼素数的工作一直稳步发展。到 1998 末，已知 37 个不同

○ 莫孙尼 (Marin Mersenne, 1588—1648) 莫孙尼生于法国 Maine 的一个劳动家庭，上过 Mans 学院和位于 La Flèche 的耶稣学院。后来又从 1609 年到 1611 年在 Sorbonne 继续接受教育，学习神学。1611 年他加入广称为 Minims 的宗教团体。Minims 的名字来自 Minimi (极小) 一词 (这些人自认为是宗教信条最少的团体)。除祈祷外，他们献身于学术和研究。1612 年他在巴黎的 Place Royale 当了牧师。1614 年至 1618 年之间在 Nevers 的 Minims 女修道院教哲学。1619 年回到巴黎。随后他在 de l'Annociade 的 Minims 宿舍成了包括 Fermat 和 Pascal 在内的法国科学家、哲学家和数学家的聚会场所。莫孙尼与全欧洲的学者频繁通信，作为数学和科学知识的交流者，起着数学学术杂志的作用 (今天的因特网也起这种作用)。莫孙尼写的书内容广泛，包括力学、数学物理、数学、音乐和声学。他研究素数，并尝试构造一个能表示所有素数的公式，但没有成功。1614 年莫孙尼声称，对 $p=2$、3、5、7、13、17、19、31、67、127、257，2^p-1 是素数，而对于小于 257 的所有其他素数，2^p-1 不是素数。人们花了 300 多年的时间在莫孙尼的上述论断中找到 5 个错误。具体来说，对 $p=67$ 和 $p=257$，2^p-1 不是素数，而对 $p=61$，$p=87$ 和 $p=107$，2^p-1 是素数。还值得一提的是，莫孙尼为受到宗教批评的两位当时最出名的人辩护，这两个人就是笛卡儿和伽利略。他还协助揭露炼丹术士和占星术士的骗术。

的莫孙尼素数，其中6个是1990年以来找到的。最大的已知莫孙尼素数（至1998年末）是 $2^{3021377}-1$，这是个909 526位的数。作为一种集体的努力，一个称为"大因特网莫孙尼素数搜索"（GIMPS）的组织已经成立，以共同寻找新的莫孙尼素数。顺便说一句，甚至对莫孙尼素数的寻找这件事本身都有实际的意义。对超级计算机的一种质量控制测试就是重复确定一个大莫孙尼素数的卢卡斯—来赫莫测试。

由定理3知可用试除法写出一个因数分解和测试是否为素数的过程。不过这样的过程不是有效的算法；人们已经开发了许多更实用和更有效的算法完成这些任务。因数分解和素数测试对数论在密码学中的应用已变得很重要。这引起了人们极大的兴趣来开发完成这两个任务的有效算法。在过去的25年中已经研究设计了一些聪明的过程，能有效地生成大素数。然而在这同一段时间中，尽管已开发了有力的因数分解新方法，为大整数分解因数仍然要花费特别长的时间。不过分解大整数为素数的挑战引起了许多人的兴趣。因特网上有集体的努力来分解大整数，特别是形如 $k^n\pm1$ 的大数，其中 k 是小正整数而 n 是大正整数（这种数称为卡宁汉数）。任何时候，总有称为"10个最热"的这种大整数等待分解。

2.3.4 除法算法

我们已经看到，一个整数也许能也许不能被另一个整数整除。但是当一个整数被一个正整数除时，总有一个商和一个余数。下面的算法将说明这一点。

定理4 除法算法令 a 为整数，d 为正整数，那么有唯一的整数 q 和 r，其中 $0\leqslant r<d$，使得 $a=dq+r$。

注意 定理4实际上并不是一个算法。(为什么不是?) 我们采用的是它的传统称呼。

定义3 在上述除法算法给出的等式中，d 称为除数，a 称为被除数，q 称为商，r 称为余数。

用下面两个例子解释除法算法。

例8 101被11除时商和余数是什么？

解 我们有

$$101=11\cdot9+2$$

所以101被11除时商是9，余数是2。 ■

例9 -11 被3除时商和余数是什么？

解 我们有

$$-11=3\cdot(-4)+1$$

所以 -11 被3除时商是 -4，余数是1。 ■

注意整数 a 能被整数 d 整除的充分必要条件是当 a 被 d 除时余数为0。

2.3.5 最大公约数和最小公倍数

能整除两个整数的最大整数称为这两个整数的**最大公约数**。

定义 4　令 a 和 b 是不全为 0 的两个整数。能使 $d|a$ 和 $d|b$ 的最大整数 d 称为 a 和 b 的最大公约数。a 和 b 的最大公约数用 $\gcd(a,b)$ 表示。

不全为 0 的两个整数的最大公约数的确存在，因为这两个整数的公约数集合是有限的，求两个整数的最大公约数的一个方法是求出两个整数的所有正的公约数，然后取其中最大的。下面的例子就是这样做的，随后会给出一个更有效的求最大公约数的方法。

例 10　24 和 36 的最大公约数是什么？

解　24 和 36 的正公约数是 1，2，3，4，6 和 12，所以 $\gcd(24,36)=12$。■

例 11　17 和 22 的最大公约数是什么？

解　17 和 22 除 1 以外没有正公约数，所以 $\gcd(17,22)=1$。■

由于往往需要指明两个整数除 1 以外没有其他正公约数，我们给出下面的定义。

定义 5　如果整数 a 和 b 的最大公约数是 1，就说它们是互素的。

例 12　从例 11 可知整数 17 和 22 是互素的，因为 $\gcd(17,22)=1$ 。■

由于经常需要指明一个整数集合中任何两个整数都没有大于 1 的正公约数，我们给出下面的定义。

定义 6　整数 $a_1,a_2,\cdots,a_n$ 称为两两互素，如果只要 $1\leqslant i\leqslant j\leqslant n$ ，就有 $\gcd(a_i,a_j)=1$ 。

例 13　判断整数 10，27 和 21 是否两两互素，整数 10，19 和 24 是否两两互素。

解　由于 $\gcd(10,17)=1,\gcd(10,21)=1$ 和 $\gcd(17,21)=1$ ，我们断定 10，17 和 21 是两两互素的。

由于 $\gcd(10,24)=2>1$ ，所以 10，17 和 24 不是两两互素的。■

求两个整数的最大公约数的另一方法是利用这两个整数的素因子分解。假定两个全不为 0 的整数 a 和 b 的素因子分解为

$$a=p_1^{a_1}p_2^{a_2}\cdots p_n^{a_n},\quad b=p_1^{b_1}p_2^{b_2}\cdots p_n^{b_n}$$

其中每个指数都是非负整数，而且出现在 a 和 b 分解中的所有素数都包含在两个分解之中，必要时以 0 为指数出现。于是 $\gcd(a,b)$ 由下面的公式给出：

$$\gcd(a,b)=p_1^{\min(a_1,b_1)}p_2^{\min(a_2,b_2)}\cdots p_n^{\min(a_n,b_n)}$$

其中 $\min(x,y)$ 代表两个数 x 和 y 的最小值。为证明这一计算 $\gcd(a,b)$ 的公式是有效的，必须证明等式右边的整数整除 a 和 b，而且没有比它大的也能整除 a 和 b 的整数。由于右边每个素数的指数都不超过 a 和 b 的分解中该素数的指数，所以它的确整除 a 和 b。进一步说，没有更大的整数能整除 a 和 b，因为分解式中每个素数的指数不能增大，而且其他素数也不能加进来。

例 14　由于 120 和 500 的素因子分解分别是 $120=2^3\cdot 3\cdot 5$ 和 $500=2^2\cdot 5^3$，所以它们的最大公约数是

$$\gcd(120,500)=2^{\min(3,2)}3^{\min(1,0)}5^{\min(1,3)}=2^2\cdot 3^0\cdot 5^1=20$$ ■

素因子分解也可用于求两个整数的最小公倍数。

定义 7 正整数 a 和 b 的最小公倍数是能被 a 和 b 整除的最小正整数。a 和 b 的最小公倍数用 $\mathrm{lcm}(a,b)$ 表示。

最小公倍数的存在是由于能被 a 和 b 整除的整数集合是非空的，而每个非空的正整数集合都有一个最小元素（根据第 3 章将讨论的良序性质）。假定 a 和 b 的素因子分解如前面给出的那样，那么 a 和 b 的最小公倍数由下式给出：

$$\mathrm{lcm}(a,b)=p_1^{\max(a_1,b_1)}p_2^{\max(a_2,b_2)}\cdots p_n^{\max(a_n,b_n)}$$

其中 $\max(x,y)$ 表示两个数 x 和 y 中的最大数。这一公式是有效的，因为 a 和 b 的公倍数在其分解中至少含 $\max(a_i,b_i)$ 个 p_i，而最小公倍数的分解中又没有 a 和 b 的因子之外的素数。

例 15 $2^3 3^5 7^2$ 和 $2^4 3^3$ 的最小公倍数是什么？

解 我们有

$$\mathrm{lcm}(2^3 3^5 7^2, 2^4 3^3)=2^{\max(2,4)}3^{\max(5,3)}7^{\max(2,0)}=2^4 3^5 7^2$$ ■

下面的定理给出了两个整数的最大公约数和最小公倍数之间的关系。用上面给出的求这两个数的公式就可以证明这一定理。具体证明留给读者作练习。

定理 5 令 a 和 b 为正整数，则

$$ab=\gcd(a,b)\cdot\mathrm{lcm}(a,b)$$

2.3.6 模运算

有时我们只关心一个整数被另一个指定的正整数除时的余数。例如问从现在开始 50 个小时以后的时间（每天 24 小时），我们关心的是当前钟点数加上 50 除以 24 的余数。由于往往只对余数感兴趣，我们用专门的符号表示它们。

定义 8 令 a 为整数，m 为正整数。用 $a \bmod m$ 表示 a 被 m 除的余数。

从余数的定义知，$a \bmod m$ 是使 $a=qm+r$ 且 $0\leqslant r<m$ 的整数 r。

例 16 我们有 $17 \bmod 5=2$，$-133 \bmod 9=2$，以及 $2001 \bmod 101=82$。 ■

还有一个用来表示两个整数被正整数 m 除时有同样余数的符号。

定义 9 若 a 和 b 为整数而 m 为正整数，如果 m 整除 $a-b$，就说 a 与模 m 同余，用 $a\equiv b \pmod m$ 表示。如果 a 和 b 不是模 m 同余的，就写成 $a\not\equiv b \pmod m$。

注意 $a\equiv b \pmod m$ 当且仅当 $a \bmod m=b \bmod m$。

例 17 判断 17 是否和 5 模 6 同余，24 是否和 14 模 6 同余。

解 由于 6 整除 $17-5=12$，所以 $17\equiv 5 \pmod 6$。但 $24-14=10$ 不能被 6 整除，所以 $24\not\equiv 14 \pmod 6$。 ■

伟大的德国数学家高斯[⊖]在 18 世纪末发展了同余的概念。

同余概念在数论发展中起着重要的作用。下面的定理给出了一个使用同余的方法。

定理 6　令 m 为正整数。整数 a 和 b 模 m 同余的充分必要条件是存在整数 k，使 $a=b+km$。

证　若 $a\equiv b\ (\bmod\ m)$，那么 $m\mid(a-b)$。这表示有整数 k 使 $a-b=km$，于是 $a=b+km$。反过来，若有整数 k 使 $a=b+km$，那么 $km=a-b$。于是 m 整除 $a-b$，所以 $a\equiv b\ (\bmod\ m)$。 □

下面的定理说明同余和加法乘法的关系。

定理 7　令 m 为正整数。若 $a\equiv b(\bmod\ m)$，$c\equiv(\bmod\ m)$，那么 $a+b\equiv b+d(\bmod\ m)$ 及 $ac=bd(\bmod\ m)$。

证　因为 $a\equiv b(\bmod\ m)$，$c\equiv b(\bmod\ m)$，所以有整数 s 和 t 使 $b=a+sm$ 和 $d=c+tm$。于是

$$b+d=(a+sm)+(c+tm)=(a+c)+(s+t)m$$

及

$$bd=(a+sm)(c+tm)=ac+m(at+cs+stm)$$

因此

$$a+c\equiv b+d(\bmod\ m)\text{及}\ ac\equiv bd(\bmod\ m)$$ □

例 18　由于 $7\equiv 2\ (\bmod\ 5)$ 和 $11\equiv 1\ (\bmod\ 5)$，从定理 7 知

$$18=7+11\equiv 2+1=3(\bmod\ 5)$$

且

$$77=7\cdot 11\equiv 2\cdot 1=2(\bmod\ 5)$$ ■

2.3.7　同余应用

数论有广泛的应用。本节介绍其中三个：用同余为计算机文件分配内存地址，生成伪随机数，以及基于模运算的密码系统。

⊖ 高斯(Karl Friedrich Gauss，1777—1855)　泥瓦匠的儿子高斯是个神童，10 岁时即展现了非凡的潜力，迅速解答了老师出的难题。当时老师要学生求头 100 个正整数的和。高斯发现这 100 个数的和可以通过 50 对和求出，每一对都是 101：$100+1$，$99+2$，…，$50+51$。智慧的火花引起了包括布朗斯威克的费迪南德公爵在内的赞助人的注意。在他们的提议下，高斯得以在卡洛林学院和哥廷根大学学习。高斯还是个学生，就发明了从实验结果估计变量最可能的值的最小平方方法。1796 年高斯作出了几何学上的基本发现，推动了自古代即已停步不前的这一学科的发展。他证明了只用圆规和直尺可以画正 17 边形。

1799 年高斯给出了算术基本定理第一个严格的证明。这一定理指出 n 次多项式恰有 n 个根（计算重根）。当他成功地用不充分的数据计算出人类首次发现的小行星谷神星的轨道时，高斯赢得了世界声誉。

高斯被他同时代的数学家称为数学王子。尽管高斯以其在几何、代数、数学分析、天文和物理上的许多发现而知名，他却对数论有着特别的兴趣。从他的名言可见一斑：“数学是科学的皇后，而数论是数学的皇后。”1801 年高斯出版了《*Disquisitiones Arithmeticae*》一书，为现代数论奠定了基础。

例 19　**散列函数**　学校的中央计算机保存着所有学生的档案记录。怎样分配内存地址才能迅速检索到学生记录？适当选一个散列函数就是这一问题的解。记录用关键码识别，每个关键码唯一地识别一个学生记录。例如往往用学生的社会安全号作为他的记录的关键码。散列函数 h 将内存地址 $h(k)$ 分配给以 k 为关键码的记录。■

实践中使用许多不同的散列函数。最常用的散列函数之一是

$$h(k) = k \bmod m$$

其中 m 是可供使用的内存地址的数目。

散列函数应该易于计算以便快速检索到文件。散列函数 $h(k) = k \bmod m$ 符合这一要求；求 $h(k)$ 只须计算 k 被 m 除的余数。散列函数还应该是映上的，这样所有内存地址均可利用。函数 $h(k) = k \bmod m$ 也符合这一要求。

例如，当 $m = 111$ 时，以 064212848 为社会安全号的学生的记录分配到的地址是 14，因为

$$h(064212848) = 064212848 \bmod 111 = 14$$

类似地，由于

$$h(037149212) = 037149212 \bmod 111 = 65$$

以 037149212 为社会安全号的学生的记录得到的内存地址是 65。

由于散列函数不是一对一的（因为关键码的个数可能大于内存地址数），有可能多个记录分配到同一个内存地址。这时我们说出现了冲突。消除冲突的一个办法是使用散列函数给出的但已被占用的地址后面第一个未占用地址。例如在分配了上述两个地址以后，我们把 15 分配给社会安全号为 107405723 的学生记录。要看清这一点，首先注意 h（k）会把这一社会安全号映射到地址 14，这是因为

$$h(107405723) = 107405723 \bmod 111 = 14$$

但这一地址已被占用（社会安全号为 064212848 的学生文件占用），而 15 是内存地址 14 后面第一个未占用的地址。

有许多较上述简单方法复杂的消除冲突的办法。本书结束处给出的散列函数参考文献讨论了这些方法。

例 20　**伪随机数**　计算机模拟常需要随机选择的数目。有不同的方法可以产生具有随机选取性质的数。由于用系统的方法产生的数不可能真正是随机的，就称为伪随机数。■

最常用的产生伪随机数的过程称为线性同余法。我们选择 4 个数：模数 m，乘数 a，增量 c 和种数 x_0，使 $2 \leqslant a < m$，$0 \leqslant c < m$，及 $0 \leqslant x_0 < m$。我们生成一个伪随机数序列 $\{x_n\}$ 使得对所有 n，$0 \leqslant x_n < m$。生成的办法是逐次同余：

$$x_n + 1 = (ax_n + c) \bmod m$$

（这是个递归定义的例子，递归定义将在 3.3 节讨论。在这一节只证明这样定义序列是有效的。）

不少计算机试验都要求产生 0 和 1 之间的伪随机数。要得到这样的数目，可以用模数除线性同余生成器产生的数，即使用 x_n/m。

例如选 $m=9$，$a=7$，$c=4$ 和 $x_0=3$，产生的伪随机数序列如下：

$$x_1=7x_0+4=7\cdot 3+4=25 \bmod 9=7$$
$$x_2=7x_1+4=7\cdot 7+4=53 \bmod 9=8$$
$$x_3=7x_2+4=7\cdot 8+4=60 \bmod 9=6$$
$$x_4=7x_3+4=7\cdot 6+4=46 \bmod 9=1$$
$$x_5=7x_4+4=7\cdot 1+4=11 \bmod 9=2$$
$$x_6=7x_5+4=7\cdot 2+4=18 \bmod 9=0$$
$$x_7=7x_6+4=7\cdot 0+4=4 \bmod 9=4$$
$$x_8=7x_7+4=7\cdot 4+4=32 \bmod 9=5$$
$$x_9=7x_8+4=7\cdot 5+4=39 \bmod 9=3$$

由于 $x_9=x_0$ 而且每一项都只依赖于其前面一项，所以产生的序列如下：

3，7，8，6，1，2，0，4，5，3，7，8，6，1，2，0，4，5，3，…

这个序列含 9 个不同的数，然后重复。

大部分计算机的确使用线性同余生成器产生伪随机数。常使用的线性同余生成器的增量 $c=0$。这样的生成器称为纯乘式生成器。例如以 $2^{31}-1$ 为模，以 $7^5=16\ 807$ 为乘数的纯乘式生成器就广为采用。可以证明以这些值来计算，会产生 $2^{31}-2$ 个伪随机数，然后开始重复。

2.3.8 密码学

同余在离散数学和计算机科学中有许多应用。可以在本书末给出的推荐读物中找到对这些应用的讨论。最重要的同余应用之一涉及研究信息保密的密码学。已知最早使用密码的例子之一是凯撒。他把每个字母在字母表中往下移动 3 位以获得保密信息（字母表的后 3 个字母移为最前 3 个）。例如根据这一方案，字母 B 移到 E，而字母 X 移到 A。这是加密的一个例子，就是产生保密信息的过程。

要用数学来表达凯撒的加密过程，我们按照字母在字母表中的位置用 0 到 25 的数表示字母。例如用 0 表示 A，用 10 表示 K，用 25 表示 Z。凯撒的加密方法可以用函数 f 表示，对每个非负整数 p，$p\leqslant 25$，函数值 $f(p)$ 是 $\{0, 1, 2, \cdots, 25\}$ 中的一个数，使

$$f(p)=(p+3) \bmod 26$$

在加密信息中，由 p 代表的字母用 $(p+3) \bmod 26$ 代表的字母代替。

例 21 从信息 “MEET YOU IN THE PARK” 用凯撒密码获得的加密信息是什么？

解 首先用数代替信息中的字母，得

12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10

再用 $f(p)=(p+3) \bmod 26$ 代替 p，得

15 7 7 22　1 17 23　11 16　22 10 7　18 3 20 13

翻译成字母即得加密信息“PHHW BRXLQ WHK SDUN”。■

要把凯撒密码加密的保密信息还原为原来的信息，需使用 f 的反函数 f^{-1}。注意 f^{-1}把 $\{0, 1, 2, \cdots, 25\}$ 中的整数 p 变成$f^{-1}(p)=(p-3)\bmod 26$ 。换名话说，要找回原信息，每个字母要在字母表中向上移 3 位，最前 3 个字母移到最后 3 位。从加密信息恢复成原信息的过程称为解码。

有各种方法可以推广凯撒密码。例如不把每个字位移动 3 位，而是移动 k 位，于是

$$f(p)=(p+k)\bmod 26$$

这样的密码称为移位密码。注意解码可以用

$$f^{-1}(p)=(p-k)\bmod 26$$

来完成。

显然凯撒的方法和移位密码不能提供高度安全。有各种改进这一方法的办法。稍稍提高一点安全性的一种办法是使用形为

$$f(p)=(ap+b)\bmod 26$$

的函数，其中 a 和 b 为整数，而且需保证 f 为双射。（这样的映射称为仿射变换。）这种函数产生若干可能的加密系统。下面的例子解释如何使用一个这样的系统。

例 22　若用$f(p)=(7p+3)\bmod 26$ 加密用什么字母取代字母 K？

解　首先注意 10 代表 K。然后用指定的加密函数得到$f(10)=(7\cdot 10+3)\bmod 26=21$ 。因为 21 代表 V，在加密信息中用 V 代表字母 K。■

凯撒的加密法及其推广都是用字母表中的一个字母代替另一个。这类加密法很容易受到根据字母在信息中出现频率作出的攻击。复杂一些的加密法是用一段字母取代另一段字母。已经有一些根据模运算设计的加密字母段的技术。对这些技术的讨论可以在书末列出的推荐读物中找到。

练习

1. 17 能整除下列各数吗？
   a） 68　　b） 84　　c） 357　　d） 1001
2. 如果 a 是不为 0 的整数，证明：
   a） 1 整除 a。　　b） a 整除 0。
3. 证明定理 1 的第 2 部分成立。
4. 证明定理 1 的第 3 部分成立。
5. 证明如果 $a\mid b$ 及 $b\mid a$，其中 a 和 b 为整数，则必有 $a=b$ 或 $a=-b$。
6. 证明若 a，b，c，d 为整数，使 $a\mid c$ 及 $b\mid d$，则 $ab\mid cd$。
7. 若 a，b，c 为整数，使 $ac\mid bc$，求证 $a\mid b$。
8. 下列整数是素数吗？
   a） 19　　b） 27　　c） 93　　d） 101　　e） 107　　f） 113

9. 求下列各种情况的商和余数。

a) 19 被 7 除　　b) −111 被 11 除

c) 789 被 23 除　　d) 1001 被 13 除

e) 0 被 19 除　　f) 3 被 5 除

g) −1 被 3 除　　h) 4 被 1 除

10. 求下列各数的素因式分解。

a) 39　b) 81　c) 101　d) 143　e) 289　f) 899

11. 求 10! 的素数分解。

*12. 100! 的尾部有多少个 0?

*13. 不是两个整数的比的实数称为无理数。求证 $\log_2 3$ 是个无理数。

14. 小于 12 的哪些正整数与 12 互素?

15. 判断下列各组整数是否两两互素?

a) (11, 15, 19)　　b) (14, 15, 21)

c) (12, 17, 31, 37)　　d) (7, 8, 9, 11)

16. 如果一个正整数等于除它自己以外它的所有正因子的和，这个数就称为完全数。

a) 证明 6 和 28 是完全数。

b) 当 2^p-1 为素数时，求证 $2^{p-1}(2^p-1)$ 是完全数。

17. 令 m 为正整数。求证若 $a \bmod m = b \bmod m$，则 $a \equiv b \pmod m$。

18. 令 m 为正整数。求证若 $a \equiv b \pmod m$，则 $a \bmod m = b \bmod m$。

19. 求证若 2^n-1 为素数，则 n 为素数。[提示：利用等式$2^{ab}-1=(2^a-1)\cdot(2^{a(b-1)}+2^{a(b-2)}+\cdots+2^a+1)$。]

20. 判断下列各整数是否为素数，以此证明莫孙尼的某些论断。

a) 2^7-1　b) 2^9-1　c) $2^{11}-1$　d) $2^{13}-1$

21. 欧拉 ϕ 函数在正整数 n 的值定义为小于或等于 n 且与 n 互素的正整数的个数。(注意 ϕ 是希腊字母。) 求

a) $\phi(4)$　b) $\phi(10)$　c) $\phi(13)$

22. 求证 n 为素数的充分必要条件是$\phi(n)=n-1$。

23. $\phi(p^k)$值是什么? 其中 p 为素数，k 为正整数。

24. 下列各对整数的最大公约数是什么?

a) $2^2\cdot3^3\cdot5^5$, $2^5\cdot3^3\cdot5^2$

b) $2\cdot3\cdot5\cdot7\cdot11\cdot13$, $2^{11}\cdot3^9\cdot11\cdot17^{14}$

c) 17, 17^{17}

d) $2^2\cdot7$, $5^3\cdot13$

e) 0, 5

f) $2\cdot3\cdot5\cdot7$, $2\cdot3\cdot5\cdot7$

*25. 若 n 和 k 为正整数，求证$\lceil n/k\rceil=\lfloor(n-1)/k\rfloor+1$。

26. 若 a 为整数，d 是大于 1 的正整数，求证 a 除以 d 的商和余数分别是$\lceil a/d\rceil$和 $a-d\lfloor a/d\rfloor$。

27. 设 m 为正整数，给出计算与整数 a 模 m 同余的绝对值最小的整数的公式。

28. 计算下列各量：

a) −17 mod 2　　b) 144 mod 7
c) −101 mod 13　　d) 199 mod 19

29. 计算下列各量：
a) 13 mod 3　　b) −97 mod 11
c) 155 mod 19　　d) −221 mod 23

30. 列出 5 个与 4 模 12 同余的整数。

31. 判断下列各整数是否与 5 模 17 同余。
a) 80　　b) 103　　c) −29　　d) −122

32. 若两个整数的乘积为 $2^7 3^8 5^2 7^{11}$，它们的最大公约数为 $2^3 3^4 5$，它们的最小公倍数是什么？

33. 若 a 和 b 为正整数，求证 $ab = \gcd(a,b) \cdot \mathrm{lcm}(a,b)$。[提示：利用 a 和 b 的素因子分解以及根据素因子分解给出的计算 $\gcd(a,b)$ 和 $\mathrm{lcm}(a,b)$ 的公式。]

34. 若 $a \equiv b \pmod m$，$c \equiv d \pmod m$，其中 a，b，c，d，m 为整数，且 $m \geq 2$，求证 $a - c \equiv b - d \pmod m$。

35. 若 $n \mid m$，m 为大于 1 的正整数，且 $a \equiv b \pmod m$，其中 a, b 为整数，求证 $a \equiv b \pmod n$。

36. 求证若 a，b，c，m 为整数，$m \geq 2$，$c > 0$ 且 $a \equiv b \pmod m$，则 $ac \equiv bc \pmod{mc}$。

37. 若 a，b，c，m 为整数，$m \geq 2$，证明 $ac \equiv bc \pmod m$ 不一定蕴含 $a \equiv b \pmod m$。

38. 若 a，b，m 为整数，$m \geq 2$ 且 $a \equiv b \pmod m$，求证 $\gcd(a, m) = \gcd(b, m)$。

39. 若 a，b，k，m 为整数，$k \geq 1$，$m \geq 2$ 且 $a \equiv b \pmod m$，求证 $a^k \equiv b^k \pmod m$。

40. 若用散列函数 $h(k) = k \bmod 101$ 和社会安全号为学生记录分配地址，下列社会安全号得到的地址是什么？
a) 104578690　　b) 432222187
c) 372201919　　d) 501338753

41. 停车场有 31 个车位供来访者使用，编号为 0 到 30。来访者根据散列函数 $h(k) = k \bmod 31$ 以及车牌前 3 位数获得车位。
a) 下列车牌前三位数得到什么车位？
317，918，007，100，111，310
b) 给出一个办法，使来访者在发现他们得到的车位已被占用时可以找到空车位。

42. 用线性同余生成器 $x_{n+1} = (4x_n + 1) \bmod 7$ 和种子数 $x_0 = 3$ 产生的伪随机数序列是什么？

43. 用纯乘式生成器 $x_{n+1} = 3x_n \bmod 11$ 和种子数 $x_0 = 2$ 产生的伪随机数序列是什么？

44. 写一伪码算法，用线性同余生成器产生伪随机数序列。

45. 为信息“DO NOT PASS GO”加密，先把字母翻译成整数，再用下面给的加密函数计算，然后把整数翻译成字母。
a) $f(p) = (p + 3) \bmod 26$（凯撒密码）
b) $f(p) = (p + 13) \bmod 26$
c) $f(p) = (3p + 7) \bmod 26$

46. 为下列用凯撒密码加密的信息解密。
a) EOXH MHDQV
b) WHVW WRGDB

c) HDW GLP VXP

书籍用国际标准书号（ISBN）来识别，这是个十位数字编码 $x_1x_2\cdots x_{10}$，由出版者给定。这 10 位码分段表示语言、出版者、出版社给书的编号，以及最后一个校验位。校验位或是数字，或是字母 X（用来表示 10)。校验位用 $\sum_{i=1}^{10} ix_i \equiv 0 (\mathrm{mod}\ 11)$ 来选择，用于发现各位数字的错误或数字移位。

47. 本书第 3 版的 ISBN 前九位数字是0-07-053965，它的校验位是什么？
48. 《*Elementary Number Theory and Its Applications*》第 3 版的 ISBN 是0-201-57Q89-1，其中 Q 是个数字。求 Q 的值。
49. 判断出版者给出的本书 ISBN 的校验位是否计算正确。
50. 求恰有 n 个不同因数的最小正整数，其中，n 是

    a) 3　　b) 4　　c) 5　　d) 6　　e) 10
51. 给出与素数或素数分解有关的公式或规则，用以计算序列的第 n 项，使序列的初始项为下面给出的这些。

    a) 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, …
    b) 1, 2, 3, 2, 5, 2, 7, 2, 3, 2, 11, 2, 13, 2, …
    c) 1, 2, 2, 3, 2, 4, 2, 4, 3, 4, 2, 6, 2, 4, …
    d) 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, …
    e) 1, 2, 3, 3, 5, 5, 7, 7, 7, 7, 11, 11, 13, 13, …
    f) 1, 2, 6, 30, 210, 2310, 30030, 510510, 9699690, 223092870, …
52. 给出与素数或素数分解有关的公式或规则，用以计算序列第 n 项，使序列的初始项为下面给出的这些。

    a) 2, 2, 3, 5, 5, 7, 7, 11, 11, 11, 11, 13, 13, …
    b) 0, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, …
    c) 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1,
    d) 1, −1, −1, 0, −1, 1, −1, 0, 0, 1, −1, 0, −1, 1, 1, …
    e) 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, …
    f) 4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, …

2.4 整数和算法

2.4.1 引言

正如 2.1 节所说，算法这一术语最初指的是用整数的十进制表示进行算术运算的过程。修改后能处理二进制表示的这些算法是计算机算术运算的基础。它们为理解算法这一概念及算法复杂性提供了很好的实例。因此本节将讨论这些算法。

除算术中常用的整数算法以外，还有许多重要的算法涉及整数。我们将从欧几里德算法开始我们的讨论。这是最有用的算法之一，很可能还是数学中最古老的算法。我们还将描述求解一个整数的基数 b 展开的算法，其中 b 是任意基数。

2.4.2 欧几里德算法

2.3节给出的用整数的素因子分解求两个整数最大公约数的算法效率不高。原因是求素因子分解消耗太多时间。我们将给出一个称为欧几里德算法的效率更高的方法求最大公约数。这个算法古代就有了。这是用古希腊数学家欧几里德[⊖]的名字命名的，因为在他的书《*Elements*》中记载了这一算法。

在介绍欧几里德算法之前，我们先看看它怎样求gcd(91,287)。首先用两数中的小者91去除两数中的大者287，得到

$$287 = 91 \cdot 3 + 14$$

91和287的任何公约数必定也是 $287 - 93 \cdot 3 = 14$ 的因数。而且91和14的任何公约数也必定是 $287 = 93 \cdot 3 + 14$ 的因数。因此，287和91的最大公约数和91与14的最大公约数相同。这表明，求gcd(91,287)的问题已被化简为求gcd(91,14)的问题。

下一步用14除91得

$$91 = 14.6 + 7$$

由于91和14的任何公约数也能整除 $91 - 14 \cdot 6 = 7$，同时14和7的任何公约数整除91，所以 $\gcd(91,14) = \gcd(14,7)$。

继续用7除14，得

$$14 = 7 \cdot 2$$

因为7整除14，得 $\gcd(14,7) = 7$，同时由于 $\gcd(287,91) = \gcd(91,14) = \gcd(14,7) = 7$，最初的问题得解。

现在我们介绍欧几里德算法在一般情况下如何工作。我们将用辗转相除把求两个正整数最大公约数的问题化简为求两个较小整数的最大公约数，直到两个整数中的一个为0。

欧几里德算法的基础是关于最大公约数的下述结论和除法算法。

引理1 令 $a = bq + r$，其中 a，b，q，r 为整数，则 $\gcd(a,b) = \gcd(b,r)$。

证 如果我们能证明 a 与 b 的公约数和 b 与 r 的公约数相同，也就证明了 $\gcd(a,b) = \gcd(b,r)$，因为这两对整数必定有相同的最大公约数。

现在假定 d 整除 a 和 b。于是 d 也整除 $a - bq = r$（根据2.3节定理1）。因此 a 和 b 的任何公约数也是 b 和 r 的公约数。

类似地，假定 d 整除 b 和 r。于是 d 也整除 $bq + r = a$。因此 b 和 r 的任何公约数也是 a 和 b 的公约数。

于是 $\gcd(a,b) = \gcd(b,r)$。 □

⊖ 欧几里德（Euclid，大约公元前350年） 欧几里德是人类最成功的数学著作《*Elements*》的作者，这本书自古至今已有1000多个不同的版本。人们对欧几里德的生平所知甚少，只知道他曾在著名的亚历山大学院任教。显然欧几里德不强调应用。当学生问他学了几何有什么用时，他解释说知识本身就值得学习，并让仆人给了该学生一枚硬币，"因为他一定要从学习中获利"。

假定 a 和 b 为正整数，$a \geq b$。令 $r_0 = a$，$r_1 = b$。若辗转应用除法算法，得

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2 & 0 \leq r_2 < r_1 \\
r_1 &= r_2 q_2 + r_3 & 0 \leq r_3 < r_2 \\
&\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n & 0 \leq r_n < r_{n-1} \\
r_{n-1} &= r_n q_n
\end{aligned}
$$

最终在辗转相除序列中会出现余数为0，因为在余数序列 $a = r_0 > r_1 > r_2 > \cdots \geq 0$ 中至多包含 a 项。进而从引理1知

$$
\begin{aligned}
\gcd(a,b) &= \gcd(r_0,r_1) = \gcd(r_1,r_2) = \cdots = \gcd(r_{n-2},r_{n-1}) \\
&= \gcd(r_{n-1},r_n) = \gcd(r_n,0) = r_n
\end{aligned}
$$

因此，最大公约数是除法序列中最后一个非零余数。

例 1 用欧几里德算法求414和662的最大公约数。

解 辗转使用除法算法给出：

$662 = 414 \cdot 1 + 248$

$414 = 248 \cdot 1 + 166$

$248 = 166 \cdot 1 + 82$

$66 = 82 \cdot 2 + 2$

$82 = 2 \cdot 41$

因此，$\gcd(414,662) = 2$，因为2是最后一个非零余数。 ■

算法1是用伪码写成的欧几里德算法。

算法 1 欧几里德算法

```
Procedure gcd (a,b: 正整数)
x := a
y := b
while y≠0
begin
    r := x mod y
    x := y
    y := r
end {gcd(a,b)是 x}
```

算法1中 x 和 y 的初值分别是 a 和 b。在过程的每一步都是 x 用 y 代替，而 y 用 $x \bmod y$ 代替，$x \bmod y$ 即是 x 被 y 除的余数。只要 $y \neq 0$，这个过程就重复下去。当 $y = 0$ 时算法终止，此时 x 的值，也就是这一过程中最后一个非零余数，即为 a 和 b 的最大公约数。

我们将在3.3节研究欧几里德算法的时间复杂性，并证明求 a 和 b 的最大公约数需要做

的除法次数在 $a \geq b$ 时为 $O(\log b)$。

2.4.3 整数表示

日常生活中都用十进制符号表示整数。例如，965 用来表示 $9 \cdot 10^2 + 6 \cdot 10 + 5$。不过有时用 10 以外的数做基数更方便。特别是计算机通常用二进制符号（以 2 为基数）来做算术运算，而用八进制（基数为 8）或十六进制（基数为 16）符号来表示字符，如字母或数字。事实上我们可以用 1 以外的任何正整数为基数来表示整数。下面的定理陈述的就是这一结论。

定理 1　令 b 为大于 1 的正整数。那么如果 n 是个正整数，就可以唯一地表示为下面的形式：

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

其中 k 是个非负整数，a_0，$a_1 \cdots a_k$ 是小于 b 的非负整数，$a_k \neq 0$。

此定理的证明可以在书末列出的推荐读物中找到。定理 1 中给出的 n 的表示称为 n 的以 b 为基数的展开。n 的基数 b 展开表示为 $(a_k a_{k-1} \cdots a_1 a_b)$。例如，$(245)_8$ 表示 $2 \cdot 8^2 + 4 \cdot 8 + 5 = 165$。

以 2 为基数就得到整数的二进制展开。在二进制符号中每位数字或是 0，或是 1。换言之，整数的二进制展开就是位串。计算机用二进制展开（及作为二进制展开变种的相关展开）表示整数并做整数算术运算。

例 2　以 $(101011111)_2$ 为二进制展开的整数，其十进制展开是什么？

解　我们有

$$(101011111)_2 = 2^8 + 2^6 + 2^4 + 2^3 + 2^2 + 2 + 1 = 351$$

■

16 是计算机科学中使用的另一基数。整数的基数 16 展开称为十六进制展开。这种展开要用 16 个不同的数字。通常使用的数字是 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E 和 F，其中 A 到 F 表示的数字对应于（十进制的）10 到 15。

例 3　十六进制展开 $(2AE0B)_{16}$ 的十进制展开是什么？

解　我们有

$$\begin{aligned}(2AE0B)_{16} &= 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16 + 11 \\ &= (175627)_{10}\end{aligned}$$

■

由于十六制进数字用 4 个字位表示，而字节是长度为 8 的位串，所以字节可以用两个十六进制数字表示。例如，$(11100101)_2 = (E5)_{16}$，因为 $(1110)_2 = (E)_{16}$ 而 $(0101)_2 = (5)_{16}$。

现在我们介绍一个构造整数 n 的基数 b 展开的算法。首先，用 b 除 n 得到商和余数，即

$$n = bq_0 + a_0, \quad 0 \leq a_0 < b$$

余数 a_0 就是 n 的基数 b 展开的最右边一位数字。下一步用 b 除 q_0 得

$$q_0 = bq_1 + a_1,\ 0 \leqslant a_1 < b$$

可以看出 a_1 是 n 的基数 b 展开中右数第二个数字。继续这一过程，不断用 b 除商并以余数为新的基数 b 数字。这一过程在商为 0 时终止。

例 4 求 $(12345)_{10}$的基数 8 展开。

解 首先用 8 除 12345，得

$$12345 = 8 \cdot 1543 + 1$$

相继用 8 除商，得

$$1543 = 8 \cdot 192 + 7$$
$$192 = 8 \cdot 24 + 0$$
$$24 = 8 \cdot 3 + 0$$
$$3 = 8 \cdot 0 + 3$$

由于这些余数就是 12345 的基数 8 展开中的数字，于是

$$(2345)_{10} = (30071)_8$$

■

算法 2 中给出的伪码用于计算整数 n 的基数 b 展开 $(a_{k-1}\cdots a_1 a_0)_b$。

算法 2 构造基数 b 展开

```
Procedure base b expansion (n: 正整数)
q := n
k := 0
while q≠0
begin
    a_k := q mod b
    q := ⌊q/b⌋
    k := k + 1
end {n 的基数 b 展开是 (a_{k-1}…a_1 a_0)_b}
```

在算法 2 中，q 表示不断用 b 去除时得到的商，初值 $q = n$。基数 b 展开中的数字就是做这些除法时得到的余数，由 $q \bmod b$ 给出。在得到的商 $q = 0$ 时，算法结束。

2.4.4 整数运算算法

用整数的二进制展开作运算在计算机科学中分外重要。我们将介绍对两个表示为二进制展开的整数作加法和乘法的算法，还要以使用的字位运算的实际次数来分析这些算法的计算复杂性。在整个讨论中假定 a 和 b 的二进制展开为

$$a = (a_{n-1}a_{n-2}\cdots a_1 a_0)_2, b = (b_{n-1}b_{n-2}\cdots b_1 b_0)_2$$

从而 a 和 b 各有 n 个字位（必要时让其中一个的开头几个字位为 0）。

考虑两个用二进制符号表示的整数相加的问题。作加法的过程可以根据通常借助纸和笔做加法的办法来设计，也就是一对一对二进制数字相加，有进位时再加上进位。现在来详细描述这个过程。

要把 a 和 b 相加，首先把它们最右边的数字相加。这样可得

$$a_0 + b_0 = c_0 \cdot 2 + s_0$$

其中 s_0 是 $a+b$ 的二进制展开中最右边的一位数字，而 c_0 是进位，为 0 或 1。然后把下一对字位及进位相加，

$$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1$$

其中 s_1 是 $a+b$ 的二进制展开中的下一位（从右数）数字，c_1 是进位。继续这一过程，把两个二进制展开中对应的字位及进位相加，给出 $a+b$ 的二进制展开中从右数的下一位数字。最后把 a_{n-1}，b_{n-1}和 c_{n-1}相加得 $c_{n-1} \cdot 2 + s_{n-1}$。$a+b$ 的首位数字是$s_n = c_{n-1}$。这一过程产生 a 与 b 之和的二进制展开，即 $a+b=(s_n s_{n-1} \cdots s_1 s_0)_2$。

例 5　把$a=(1110)_2$ 和 $b=(1011)_2$ 相加。

解　按照算法中规定的步骤，首先注意到

$$a_0 + b_0 = 0 + 1 = 0 \cdot 2 + 1$$

所以 $c_0 = 0$ 而 $s_0 = 1$。然后，因为

$$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 \cdot 2 + 0$$

所以 $c_1 = 1$，而 $s_1 = 0$。继续做下去，

$$a_2 + b_2 + c_2 = 1 + 0 + 1 = 1 \cdot 2 + 0$$

于是 $c_2 = 1$ 而 $s_2 = 0$。最后由于

$$a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 \cdot 2 + 1$$

从而 $c_3 = 1$ 且 $s_3 = 1$。这表明 $s_4 = c_3 = 1$。因而 $s = a + b = (11001)_2$。相加的过程如图 2-2 所示。■

```
   1 1
   1 1 1 0
   1 0 1 1
 ---------
 1 1 0 0 1
```

图 2-2　$(1110)_2$ 和$(1011)_2$ 相加

用伪码给出的加法算法如下。

算法 3　整数相加

Procedure *add* (a, b：正整数)

{a 和 b 的二进制展开分别是 $(a_{n-1}a_{n-2}\cdots a_1 a_0)_2$ 和 $(b_{n-1}b_{n-2}\cdots b_1 b_0)_2$}

$c := 0$

（续）

```
for j := 0 to n - 1
begin
    d := ⌊(a_j + b_j + c)/2⌋
    s_j := a_j + b_j + c - 2d
    c := d
end
s_n := c
{和的二进制展开是(s_n s_{n-1} ··· s_0)_2}
```

下面分析算法 3 使用的字位相加的次数。

例 6 两个二进制展开中有 4 个（或少于 n 个）字位的整数相加，算法 3 使用多少次字位加法？

解 两个整数相加是相继对字位相加，再加上进位来完成的。把两个字位及进位相加需要 3 次或少于 3 次字位加法。因此需要的字位相加总数少于二进制展开中位数的 3 倍。从而算法 3 把两个 n 位整数相加需要的字位加法次数是$O(n)$。■

下面考虑两个 n 位整数a 和b 的乘法。传统的算法（使用纸和笔作乘法）是样的：根据分配律，我们看到

$$ab = a\sum_{j=0}^{n-1} b_j 2^j = \sum_{j=0}^{n-1} a(b_j 2^j)$$

我们可以用这一等式计算 ab。首先注意在 $b_j=1$ 时 $ab_j=a$，而 $b_j=0$ 时 $ab_j=0$。每当我们用 2 乘一项时，结果都是把这一项的二进制展开向左移一位并在尾部加上一个 0。因而我们可以把 ab_j 的二进制展开向左移j 位，再在尾部加上 j 个 0 来计算（ab_j）2^j。最后，把 n 个整数ab_j2^j，$j=0$，1，2，…，$n-1$，相加就得到 ab。

下面的例子解释怎样使用这一算法。

例 7 求$a=(110)_2$ 和 $b=(101)_2$ 的乘积。

解 首先注意

$$ab_0 \cdot 2^0 = (110)_2 \cdot 1 \cdot 2^0 = (110)_2$$
$$ab_1 \cdot 2^1 = (110)_2 \cdot 0 \cdot 2^1 = (0000)_2$$

及

$$ab_2 \cdot 2^2 = (110)_2 \cdot 1 \cdot 2^2 = (11000)_2$$

为求乘积，把 $(110)_2$，$(0000)_2$ 和 $(11000)_2$ 相加，完成这些加法（用算法 3，必要时加起始 0 位）即得$ab=(11110)_2$。这一过程如图 2-3 所示。■

```
      1 1 0
      1 0 1
    -------
      1 1 0
    0 0 0
  1 1 0
  ---------
  1 1 1 1 0
```

图 2-3 $(110)_2$ 和$(101)_2$ 相乘

这个乘法过程可以用伪码描述如下。

算法 4　整数相乘

Procedure *multiply* $(a, b$:正整数$)$
$\{a$ 和 b 的二进制展开分别是 $(a_{n-1}a_{n-2}\cdots a_1a_0)_2$
　和 $(b_{n-1}b_{n-2}\cdots b_1b_0)_2\}$
for $j := 0$ **to** $n-1$
begin
　　if $b_j = 1$ **then** $c_j := a$ 移 j 位
　　else $c_j := 0$
end
$\{c_0, c_1, \cdots, c_{n-1}$是部分积$\}$
$p := 0$
for $j := 0$ **to** $n-1$
　　$p := p + c_j$
$\{p$ 是 ab 的值$\}$

下面来判断算法 4 作乘法时使用的字位相加次数和移位次数。

例 8　用算法 4 计算 a 和 b 的乘积需用多少次字位的加法和移位？

解　算法 4 计算 a 和 b 乘积的办法是把部分乘积 c_0，c_1，c_2，…，c_{n-1}相加。当 $b_j = 1$ 时，部分积 c_j 的计算是移动 a 的二进制展开 j 位。当 $b_j = 0$ 时，因为 $c_j = 0$，所以不需要移位。于是，为求出所有 n 个整数 ab_j2^j，$j = 0, 1, 2, \cdots, n-1$，需要至多

$$0 + 1 + 2 + \cdots + n - 1$$

次移位。因此根据 1.8 节例 4，需要的移位数是 $O(n^2)$。

要把 ab_j 从 $j = 0$ 到 $j = n-1$ 加起来，需要做一次 n 位整数，$(n+1)$ 位整数，…，和 $2n$ 位整数的加法。从例 6 知道，这些加法都需要 $O(n)$ 次字位相加。因此完成所有 n 个数的加法需要 $O(n^2)$ 次字位加。 ■

令人吃惊的是，还有比传统的整数乘法算法更有效的算法。其中一个算法使用 $O(n^{1.585})$ 次位运算来完成 n 位数的乘法，将在第 5 章介绍。

练习

1. 用欧几里德算法求

a) gcd(12,18) b) gcd(111,201)

c) gcd(1001,1331) d) gcd(12345,54321)

2. 用欧几里德算法求

a) gcd(1,5) b) gcd(100,101)

c) gcd(123,277) d) gcd(1529,14039)

e) gcd(1529,14038) f) gcd(11111,111111)

3. 用欧几里德算法求gcd(21,34)要做多少次除法?

4. 用欧几里德算法求gcd(34,55)要做多少次除法?

5. 把下列整数从十进制表示转换为二进制表示。

a) 231 b) 4532 c) 97644

6. 把下列整数从十进制表示转换为二进制表示。

a) 321 b) 1023 c) 100632

7. 把下列整数从二进制表示转换为十进制表示。

a) 1 1111 b) 10 0000 0001

c) 1 0101 0101 d) 110 1001 0001 0000

8. 把下列整数从二进制表示转换为十进制表示。

a) 1 1011 b) 10 1011 0101

c) 11 1011 1110 d) 111 1100 0001 1111

9. 设计一个简单的方法把十六进制表示转换为二进制表示。

10. 设计一简单的方法把二进制表示转换为十六进制表示。

11. 把下列整数从十六制进表示转换为二进制表示。

a) 80E b) 135AB

c) ABBA d) DEFACED

12. 把下列整数从二进制表示转换为十六进制表示。

a) 1111 0111 b) 1010 1010 1010

c) 111 0111 0111 0111

13. 证明每个正整数都可以唯一地表示为 2 的不同乘幂的和。[提示：考虑整数的二进制展开。]

14. 可以证明每个整数都能唯一地表示为

$$e_k 3^k + e_{k-1} 3^{k-1} + \cdots + e_1 3 + e_0$$

的形式，其中 $e_j = -1$，0 或 1，$j = 0, 1, 2, \cdots, k$。这一类展开称为平衡三进制展开。求下列整数的平衡三进制展开

a) 5 b) 13 c) 37 d) 79

15. 证明正整数被 3 整除的充分必要条件是它的十进数字之和能被 3 整除。

16. 求证正整数能被 11 整除的充分必要条件是它的偶数位十进数字之和与奇数位十进数字之和的差能被 11 整数。

17. 求证整数能被 3 整除的充分必要条件是它的偶数位二进数字之和与奇数位二进数字之和的差能被 3 整除。

整数的 1 补表示可以简化计算机算术。用小于 2^n-1 的绝对值表示正负整数，共需要 n 个字位。最左边的一位表示符号。这个位置上的 0 表示正整数，1 表示负整数。对正整数而言，其他位置上就是该整数的二进制展开。对负整数来说，其他位置上是其绝对值二进制展开中各位数字的补：1 的补是 0，0 的补是 1。

18. 用长度为 6 的位串，求下列整数的 1 补表示。

    a) 22　　b) 31　　c) −7　　d) −19

19. 下列长度为 5 的 1 补表示代表哪些整数？

    a) 11001　　b) 01101　　c) 10001　　d) 11111

20. 如果 m 是小于 2^{n-1} 的正整数，若用长度为 n 的位串，怎样从 m 的 1 补表示求（$-m$）的 1 补表示？

21. 怎样从两个整数的 1 补表示计算它们的和的 1 补表示？

22. 怎样从两个整数的 1 补表示计算它们的差的 1 补表示？

23. 求证以（$a_{n-1}a_{n-2}\cdots a_1a_0$）为 1 补表示的整数 m 可以用公式 $m=-a_{n-1}(2^{n-1}-1)+\sum_{i=0}^{n-2}a_i2^i$ 计算。

整数的 2 补表示也可以简化计算机算术，而且比 1 补表示更通用。对给定的正整数 n，要表示满足 $-2^{n-1}\leqslant x\leqslant 2^{n-1}-1$ 的整数 x，共需要 n 个字位。最左边的一位表示符号。这一位置的 0 表示正整数，1 表示负整数。这和 1 补表示一样。对正整数来说，其余位置上与该整数的二进展开一样。对负整数而言，其余位置上是 $2^{n-1}-|x|$ 的二进展开。整数的 2 补表示常用于计算机运算，因为不论整数是正是负，都很容易用它们的这种表示做加法和减法。

24. 用长度为 6 的 2 补表示回答练习 18。

25. 假定每个表示是长度为 5 的 2 补表示，回答练习 19。

26. 用 2 补表示回答练习 20。

27. 用 2 补表示回答练习 21。

28. 用 2 补表示回答练习 22。

29. 证明以（$a_{n-1}a_{n-2}\cdots a_1a_0$）为 2 补表示的整数 m 可以用等式 $m=-a_{n-1}\cdot 2^{n-1}+\sum_{i=0}^{n-2}a_i2^i$ 来计算。

30. 给出一个简单的算法，从整数的 1 补表示求它的 2 补表示。

31. 有时用 4 位二进制展开表示各个十进制数字来为整数编码。这就是整数的二进制编码的十进制形式。例如，用这种方式为 791 编码得 0111 1001 0001。用这种方式为整数编码，一个 n 位十进数需要多少字位？

    形为

$$a_nn!+a_{n-1}(n-1)!+\cdots+a_22!+a_11!$$

的和称为康托展开，其中 a_i 为整数，且 $0\leqslant a_i<i$，$i=1,2,\cdots,n$。

32. 求下列各数的康托展开。

    a) 2　　b) 7　　c) 19　　d) 87　　e) 1000　　f) 1 000 000

*33．给出一个求整数的康托展开的算法。

*34．给出一个从两个整数的康托展开求它们的和的算法

35．用课文中给出的加法算法，一步一步把 $(10111)_2$ 和 $(11010)_2$ 相加。

36．用课文中给出的算法，一步一步将 $(1110)_2$ 和 $(1010)_2$ 相乘。

37．给出一个求两个二进制展开之差的算法。

38．估计计算两个二进制展开之差需要的字位运算的次数。

39．设计一个算法，从整数 a 和 b 已知的二进制展开判断是 $a>b$，$a=b$ 还是 $a<b$。

40．当整数 a 和 b 的二进制展开有 n 个数位时，按练习 39 设计的比较算法要做多少次字位运算？

41．以需要的除法次数来估计求整数 n 的基数 b 展开的算法 2 的复杂性。

2.5　数论应用

2.5.1　引言

数论有许多应用，特别是在计算机科学中。在 2.3 节已经介绍了几个这种应用，包括散列函数，伪随机数的产生和移位密码。本节继续介绍数论，给出若干关键结果和两个重要的应用：做大整数算术运算的方法和最近发明的称为公钥系统的密码系统。

2.5.2　若干有用的结果

这一节从头到尾都会用到的一个重要结果是：两个整数 a 和 b 的最大公约数可以表示为

$$sa+tb$$

的形式，其中 s 和 t 为整数。换句话说，$\gcd(a,b)$ 可以表示为 a 和 b 的整系数线性组合。例如，$\gcd(6,14)=2$，而 $2=(-2)\cdot 6+1\cdot 14$。定理 1 说的就是这一事实。

定理 1　若 a 和 b 为正整数，则存在整数 s 和 t，使 $\gcd(a,b)=sa+tb$。

我们将不对定理 1 作形式证明，而是举例介绍一种方法，可以求两个整数的线性组合，使之等于它们的最大公约数。（本节假定线性组合均以整数为系数。）这一方法是将欧几里德算法求最大公约数所做的除法依次颠倒过来。

例 1　把 $\gcd(252,198)=18$ 表示为 252 和 198 的线性组合。

解　要证明 $\gcd(252,198)=18$，欧几里德算法做下列除法：

$$252=1\cdot 198+54$$
$$198=3\cdot 54+36$$
$$54=1\cdot 36+18$$
$$36=2\cdot 18$$

用倒数第二个（正数第三个）除法，可以把 $\gcd(254,198)=18$ 表示为 54 和 36 的线性组合，即

$$18=54-1\cdot 36$$

第二个除法告诉我们：

$$36=198-3\cdot 54$$

将 36 的这一表达式代入前一等式，可以把 18 表示为 54 和 198 的线性组合，也就是

$$18 = 54 - 1\cdot 36 = 54 - 1\cdot(198 - 3\cdot 54) = 4\cdot 54 - 1\cdot 198$$

第一个除法告诉我们：

$$54 = 252 - 1\cdot 198$$

把 54 的这一表达式代入前一个等式，可以把 18 表示为 252 和 198 的线性组合。结论是

$$18 = 4\cdot(252 - 1\cdot 198) - 1\cdot 198 = 4\cdot 252 - 5\cdot 198$$

问题得解。■

例 1 中使用的方法适合于任何一对正整数。（还有更有效的方法能把 $\gcd(a, b)$ 表示为 a 和 b 的线性组合；可参考书末的文献学习这些方法。）

我们将用定理 1 来推出几个有用的结果。我们的目标之一是证明算术基本定理的一部分，这一部分断定每个正整数只有最多一个分解。我们要证明，如果一个正整数有一个素数分解，而且素数是依不减的顺序写的，则这一分解是唯一的。

首先我们需要推导一些关于整除的结果。

引理 1 如果 a，b 和 c 为正整数，使得 $\gcd(a, b) = 1$ 且 $a \mid bc$，那么 $a \mid c$。

证 由于 $\gcd(a, b) = 1$，根据定理 1 知有整数 s 和 t，使

$$s\cdot a + t\cdot b = 1$$

用 c 乘等式两边，得

$$sac + tbc = c$$

由 2.3 节定理 1，可以用最后这一等式证明 $a \mid c$。根据该定理的第二部分，$a \mid tbc$。由于 $a \mid sac$ 和 $a \mid tbc$，由同一定理的第一部分知 a 整除 $sac + tbc$，从而 $a \mid c$。这就完成了证明。■

在证明分解唯一性时，我们将使用引理 1 的下述推广。（引理 2 的证明在 3.2 节留做练习，因为用 3.2 节介绍的数学归纳法可以很容易地证明它。）

引理 2 如果 p 是素数，且 $p \mid a_1a_2\cdots a_n$，其中 a_i 为整数，则对于某个 i，$p \mid a_i$。

现在可以证明整数分解为素数的唯一性了，也就是证明每个整数至多有一种方式写成素数的乘积，其中素数按不减少的顺序出现。这是算术基本定理的一部分。我们将在 3.2 节证明另一部分，即每个整数都有素数分解。

证（正整数的素数分解的唯一性）假定正整数 n 能用两种方式写成素数的乘积，比如说 $n = p_1p_2\cdots p_s$ 和 $n = q_1q_2\cdots q_t$，其中 p_i，p_j 都是素数，而后 $p_1 \leqslant p_2 \leqslant \cdots \leqslant p_s$ 和 $q_1 \leqslant q_2 \leqslant \cdots q_t$。

现在从两个分解式中去掉全部公有的素数，得

$$p_{i_1}p_{i_2}\cdots p_{i_u} = q_{j_1}q_{j_2}\cdots q_{j_v}$$

其中任何素数都不出现在等式两边，而 u 和 v 为正整数。由引理 2 知，有某个 k 使 p_{i_1} 整除 q_{j_k}。但任何素数都不能整除另一个素数，所以不可能有这样的 q_{j_k}。这说明至多有 n 的一种分解为素数的方式。■

用引理1也可以证明同余式两边除以同一个整数的一个结果。我们已证明（2.3节定理7）可以在同余式两边乘以同一个整数。但用同一个整数去除同余式两边并不一定得到有效的同余。下面的例子说明了这一点。

例2 同余式 $14 \equiv 8 \pmod 6$ 成立，但不能两边同除以2，因为 $14/2=7$，而 $8/2=4$，但 $7 \not\equiv 4 \pmod 6$。 ■

不过用引理1可以证明同余式两边可以同除以与模数互素的整数。这就是定理2。

定理2 令 m 为正整数，a，b 和 c 为整数。如果 $ac \equiv bc \pmod m$ 且 $\gcd(c, m)=1$，那么 $a \equiv b \pmod m$。

证 由于 $ac \equiv bc \pmod m$，则 $m \mid ac-bc=c(a-b)$。由引理1及 $\gcd(c, m)=1$，知 $m \mid a-b$。于是 $a \equiv b \pmod m$。 ■

2.5.3 线性同余

线性同余指的是形为

$$ax \equiv b \pmod m$$

的同余式，其中 m 为正整数，a 和 b 为整数，x 为变量。在数论及其应用中到处可见这种同余。

怎样解线性同余式 $ax \equiv b \pmod m$ 呢？也就是说怎样求出所有满足这一同余式的值呢？我们将要介绍的一个方法要利用使 $a\bar{a} \equiv 1 \pmod m$ 成立的整数 $\bar{a}$，如果这样的 $\bar{a}$ 存在的话。这样的 $\bar{a}$ 称为 a 的模 m 逆。在 a 和 m 互素的条件下，定理3保证 a 的模 m 逆的存在。

定理3 如果 a 和 m 为互素的整数，$m>1$，则存在 a 的模 m 逆。而且这个逆模 m 是唯一的。（即有小于 m 的唯一正整数 $\bar{a}$，它是 a 的模 m 逆，且 a 的任何别的模 m 逆均和 $\bar{a}$ 模 m 同余。）

证 由定理1及 $\gcd(a, m)=1$ 知有整数 s 和 t，使

$$sa+tm=1$$

于是

$$sa+tm \equiv 1 \pmod m$$

由于 $tm \equiv 0 \pmod m$ 所以

$$sa \equiv 1 \pmod m$$

结论是 s 为 a 的模 m 逆。本节末的练习9请读者证明这一逆是模 m 唯一的。 ■

定理3的证明同时也给出了一个在 a 和 m 互素的条件下求 a 的模 m 逆的方法：求 a 和 m 的线性组合使之等于1（按欧几里德算法步骤倒过来做即可）；这一线性组合中 a 的系数就是 a 模 m 的一个逆。我们用例3来说明这一过程。

例 3 求 3 模 7 的逆。

解 由于 $\gcd(3,7)=1$，定理 3 告诉我们存在 3 模 7 的逆。若用欧几里德算法求 3 和 7 的最大公约数，算法很快结束：

$$7=2\cdot3+1$$

从这一等式看到

$$-2\cdot3+1\cdot7=1$$

这说明 -2 是 3 模 7 的一个逆。（注意，模 7 同余于 -2 的每个整数也是 3 模 7 的逆，例如 5，-9，12 等等。） ■

有了 a 模 m 的逆 $\bar{a}$ 以后，就可以很容易地解同余方程 $ax\equiv b\ (\mathrm{mod}\ m)$；只要在线性同余式两边同乘以 $\bar{a}$ 即可。例 4 说明了这一过程。

例 4 线性同余 $3x\equiv4\ (\mathrm{mod}\ 7)$的解是什么？

解 从例 3 知道 -2 是 3 模 7 的逆。在同余式两边同乘以 -2 得

$$-2\cdot3x\equiv-2\cdot4\ (\mathrm{mod}\ 7)$$

因为 $-6\equiv1(\mathrm{mod}\ 7)$且 $-8\equiv6(\mathrm{mod}\ 7)$，所以若 x 是解，必有 $x\equiv-8\equiv6(\mathrm{mod}\ 7)$。于是根据 2.3 节定理 7，我们有

$$3x\equiv3\cdot6=18\equiv4\ (\mathrm{mod}\ 7)$$

这说明所有这种 x 满足问题中的同余式。结论是，同余方程的解是使 $x\equiv6(\mathrm{mod}\ 7)$的整数，即 6，13，20，…及 -1，-8，-15，…。 ■

2.5.4 中国余数定理

许多情况都会出现线性同余系统。例如，我们稍后会看到这种系统是做大整数算术运算的一种方法的基础。甚至可以在中国和印度的古代数学著作中发现以这种系统为文字游戏。例 5 给出的就是这样的例子。

例 5 一世纪时，中国数学家孙聪问道：

*某物不知其数，三分之余二，五分之余三，七分之余二，此物几何？*这一问题可以翻译成：求同余方程组

$$x\equiv2\ (\mathrm{mod}\ 3)$$
$$x\equiv3\ (\mathrm{mod}\ 5)$$
$$x\equiv2\ (\mathrm{mod}\ 7)$$

的解。我们将在本节稍后解这一方程组，同时也就回答了孙聪的问题。

从有关线性同余系统的中国古典问题而得名的中国余数定理，可以这样叙述：只要线性同余系统的模数两两互素，则该系统有解，而且以所有模数之乘积取模，解是唯一的。

定理 4 **中国余数定理** 令 m_1，m_2，…，m_n 为两两互素的正整数，则同余方程组

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\vdots$$
$$x \equiv a_n \pmod{m_n}$$

有唯一的模 $m = m_1 m_2 \cdots m_n$ 解。(即有一个解 x，使 $0 \leqslant x < m$，且所有其他的解均与此解模 m 同余。)

证　要建立这一定理，需要证明有一个解存在，而且是模 m 唯一的。我们将给出构造这样一个解的方法以证明解的存在；对这一解的模 m 唯一性由本节末的练习 20 证明。

要构造一个适合各方程的解，首先对 $k = 1, 2, \cdots, n$，令 $M_k = m/m_k$，即 M_k 是除 m_k 以外所有模数的乘积。由于 $i \neq k$ 时，m_i 和 m_k 没有大于 1 的公因子，所以$\gcd(m_k, M_k) = 1$。从而由定理 3 知有 M_k 模 m_k 的逆，整数 y_k，使得

$$M_k y_k \equiv 1 \pmod{m_k}$$

要得到适合所有方程的解，令

$$x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n$$

现在我们要证明 x 就是这样一个解。首先注意由于只要 $j \neq k$，就有 $M_j \equiv 0 \pmod{m_k}$，x 的和表达式中除第 k 项以外的各项模 m_k 均同余于 0。由于 $M_k y_k \equiv 1 \pmod{m_k}$，我们看到，对 $k = 1, 2, \cdots, n$，均有

$$x \equiv a_k M_k y_k \equiv a_k \pmod{m_k}$$

这就证明了 x 是这 n 个同余方程的同一解。　■

下面的例子说明怎样用定理 4 的证明中给出的构造法解同余方程组。我们将求解从例 5 中孙聪的问题产生的方程组。

例 6　要解例 5 中的同余方程组，首先令 $m = 3 \cdot 5 \cdot 7 = 105$，$M_1 = m/3 = 35$，$M_2 = m/5 = 21$，$M_3 = m/7 = 15$。我们看到，2 是 $M_1 = 35$ 的模 3 逆，因为 $35 \equiv 2 \pmod 3$；1 是 $M_2 = 21$ 的模 5 逆，因为 $21 \equiv 1 \pmod 5$；1 也是 $M_3 = 15$ 的模 7 逆，因为 $15 \equiv 1 \pmod 7$。于是这一方程组的解是那些满足下式的 x：

$$\begin{aligned} x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 &= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \pmod{105} \\ &= 233 \equiv 23 \pmod{105} \end{aligned}$$

可见 23 是所有解中最小正整数。结论是 23 是最小的正整数，被 3 除时余 2，被 5 除时余 3，被 7 除时余 2。　■

2.5.5　大整数的计算机算术运算

假定 $m_1, m_2, \cdots, m_n$ 是大于或等于 2 且两两互素的整数，令 m 为它们的乘积。根据中国余数定理可以证明(见练习 18)每个整数 a，$0 \leqslant a < m$，均可唯一地用一个 n 元组表示，这个 n 元组由 a 被 m_i 除的余数组成，$i = 1, 2, \cdots, n$。也就是说，a 可以唯一地表示为

$$(a \bmod m_1, a \bmod m_2, \cdots, a \bmod m_n)$$

例 7　求表示小于 12 的非负整数的有序偶，其中第 1 分量是用 3 除的余数，第 2 分量是用 4 除以余数。

解　求出每个整数用 3 除和用 4 除的余数，得下列表示：

$0=(0, 0)$　　$4=(1, 0)$　　$8=(2, 0)$
$1=(1, 1)$　　$5=(2, 1)$　　$9=(0, 1)$
$2=(2, 2)$　　$6=(0, 2)$　　$10=(1, 2)$
$3=(0, 3)$　　$7=(1, 3)$　　$11=(2, 3)$

■

要做大整数算术运算，我们选模数 $m_1, m_2, \cdots, m_n$，其中每个 m_i 都是大于 2 的整数，在 $i \neq j$ 时 $\gcd(m_i, m_j)=1$，且 $m=m_1 \cdot m_2 \cdots m_n$ 大于我们要做的算术运算的结果。

一旦选好了上述模数，太整数的算术运算就可以在表示它们的 n 元组的分量上做运算来完成，n 元组的分量是用大整数除以 m_i 的余数，$i=1, 2, \cdots, n$。一旦计算出表示大整数算术运算结果的 n 元组表示，就可以求解 n 个模 m_i 同余方程（$i=1, 2, \cdots, n$）找出结果的值。做大整数算术运算的这一方法有几个优点。首先可以用来完成通常一台计算机上不能做的大整数算术运算。其次，对不同模数的计算可以并行操作，加快计算速度。

例 8　假定在某台处理器上做 100 以内的整数算术运算比 100 以上的整数运算快得多，那么只要把整数表示为模两两互素的 100 以内的整数的余数的多元组，就可以将差不多所有整数计算限制在 100 以内的整数上。例如，可以以 99，98，97 和 95 为模数。（由于这些整数没有大于 1 的公因数，它们是两两互素的。）

根据中国余数定理，每个小于 $99 \cdot 98 \cdot 97 \cdot 95 = 89\ 403\ 930$ 的非负整数均可唯一地用该整数被这四个因数除的余数表示。例如把 123 684 表示为（33，8，9，89），因为 $123\ 684 \bmod 99 = 33$，$123\ 684 \bmod 98 = 8$，$123\ 684 \bmod 97 = 9$ 及 $123\ 684 \bmod 95 = 89$。类似地，413 456 可表示为（32，92，42，16）。

要求 123 684 和 413 456 的和，我们不直接用这两个数做运算，而是使用表示它们的两个 4 元组。我们把 4 元组的对应分量相加，再按相应的模数减小各个分量。这样可得

$$\begin{aligned}(33, 8, 9, 89) + (32, 92, 42, 16) &= (65 \bmod 99, 100 \bmod 98, 51 \bmod 97, 105 \bmod 95)\\ &= (65, 2, 51, 10)\end{aligned}$$

要求出和，即求出（65，2，51，10）表示的整数，必须解同余方程组

$$x \equiv 65 \pmod{99}$$
$$x \equiv 2 \pmod{98}$$
$$x \equiv 51 \pmod{97}$$
$$x \equiv 10 \pmod{95}$$

可以证明（见练习 29），537 140 是方程组唯一小于 89 403 930 的非负解，因此 537 140 是要求的和。注意只是在我们求（65，2，51，10）表示的整数时，才做大于 100 的整数的算术运算。■

大整数算术运算模数的最好选择是一组形为 2^k-1 的整数，其中 k 为正整数，这是因为很容易完成模这种整数的二进制算术，还因为容易找到两两互素的一组这种整数。（第二个理由根据的是练习 31 证明的 $\gcd(2^a-1,2^b-1)=2^{\gcd(a,b)}-1$ 这一事实。）例如，假定在我们的计算机上很容易完成 2^{35} 以内的整数算术，但更大整数的运算则要求有专门的运算过程。我们可以使用 2^{35} 以内两两互素的一组模数做整数算术运算，而这些整数可以像它们的乘积那样大。例如，练习 32 证明整数 $2^{35}-1$，$2^{34}-1$，$2^{33}-1$，$2^{31}-1$，$2^{29}-1$ 和 $2^{23}-1$ 是两两互素的。由于这 6 个模数的乘积超过 2^{184}，我们可以借助对这 6 个都不超过 2^{35} 的模数取算术模的运算完成像 2^{184} 这么大的整数的算术运算（只要运算结果也不超过这个数。）

2.5.6　伪素数

在 2.3 节我们证明了整数 n 为素数的条件是它不能被任何素数 p，$p\leqslant\sqrt{n}$，整除。不幸的是，用这一标准来证明给定的整数为素数效率不高。它要求我们找出所有不超过 $\sqrt{n}$ 的素数，还要完成用这些素数试除看是否能整除 n。

有没有效率较高的方法能判断整数是否为素数呢？古代中国数学家相信，n 为整数的充分必要条件是

$$2^{n-1}\equiv 1\ (\mathrm{mod}\ n)$$

如果这一结论成立，就可以提供一个有效的素数检测方法。为什么他们相信这一同余式能用来判断整数是否为素数呢？首先，他们观察到只要 n 为素数，这一同余一定成立。例如 5 是素数，而且

$$2^{5-1}=2^4=16\equiv 1\ (\mathrm{mod}\ 5)$$

其次，他们从未找到能使该同余式成立的合数。这些古代中国人并非全对。他们认定只要 n 是素数该同余必成立，这是对的；但他们认为只要同余成立，n 就是素数则不正确。

伟大的法国数学家费马[○]证明了当 n 为素数时该同余成立，他证明的是下述更一般性的结果。

定理 5　*费马小定理*　如果 p 为素数，a 是不能被 p 整除的整数，则

$$a^{p-1}\equiv 1\ (\mathrm{mod}\ p)$$

并且对每个整数 a，我们有

$$a^p\equiv a\ (\mathrm{mod}\ p)$$

在本节末尾的练习 17 中给出了定理 5 的证明要点。

○　费马（Pierre de Fermat，1601—1665）　费马是 17 世纪最重要的数学家之一，从职业上来说是位律师。他是历史上最著名的业余数学家。费马的数学发现发表的很少。我们从他与其他数学家的通信中了解他的工作。费马是解析几何的发明者之一，并且发展了微积分的某些基础思想。费马和帕斯卡尔（Pascal）为概率论建立了数学基础。费马提出了现在最有名的未解决的数学问题。他断定在 n 大于 2 时，方程 $x^n+y^n=z^n$ 没有非平凡的正整数解。300 多年来没有找到证明（或反例）。在他那一本古希腊数学家狄奥方托斯（Diophantus）的著作中，费马写明他有一个证明，但页边空白写不下。由于 1944 年威尔士（Andrew winles）给出的第一个证明依赖复杂的现代数学，多数人认为费马曾以为自己有过一个证明，但威尔士的证明是不正确的。不过，也许是因为自己不能给出证明，他以此诱惑别人去寻找证明。

不幸的是，有使 $2^{n-1} \equiv 1 \pmod{n}$ 的合数 n 存在。这种整数称为伪素数。

例 9　整数 341 是伪素数，因为它是合数（$341 = 11 \cdot 31$），而且在练习 23 中证明了

$$2^{340} \equiv 1 \pmod{341}$$ ■

尽管古代中国人错了，但伪素数是相对稀少的。有些更细致的检测首先就判断一个整数是否为伪素数。能通过这些检测的整数就更稀少。这种稀少性可以用做有效的概率素性检测的基础。这种检测可用来迅速证明一个已知的整数几乎肯定是素数。（更准确地说，这种检测证明的是：通过一系列检测的一个整数是素数的概率接近于 1；参看第 4 章关于概率的讨论。）这些概率素性检测能够而且已经用于在计算机上迅速寻找大素数。

2.5.7　公钥密码学

在 2.3 节我们介绍了基于同余的信息加密法。使用这些方法时，信息，也就是字符串，被译成数字。然后每个字符对应的数用移位或模 26 的仿射变换转换为另一个数。这些方法都是*私钥加密系统*的例子。知道加密密钥能使你迅速找到解密密钥。例如，用加密密钥 k 做移位密码时，代表一个字母的数目 p 发送为 $c = (p + k) \bmod 26$。解码时按 $-k$ 移位，即

$$p = (c - k) \bmod 26$$

在用私钥密码系统时，希望秘密通信的双方必须各有自己的密钥。由于知道密钥的任何人都可以轻易地为信息加密和解密，双方必须安全交换密钥。

在 20 世纪 70 年代中期，密码学者引入了*公钥密码系统*的概念。使用这种密码系统时，知道怎样给某人发信息并不能帮助你对发给此人的信息解密。在这样的一个系统中，每个人都可以有一个众所周知的加密密钥，而解密密钥是保密的，只有信息的预期接收人能解密。这是因为加密密钥并不能让人轻易找到解密密钥。只有非常大量（例如 2 亿年计算机时间）的工作才有可能解密。

1976 年，MIT 的三位研究人员——瑞弗斯特（Ron Rivest），沙米尔（Adi Shamir）和阿德来门（Len Adleman）——介绍了称为 RSA 系统的公钥密码系统，RSA 是三个发明者姓氏的首字母。RSA 密码系统依据以两个大素数的乘积为模数，对指数取模。每个人都有一个加密的密钥，密钥由两部分组成：一是模数 $n = pq$，其中 p 和 q 是约为 200 位数字的大素数；二是与（$p-1$）（$q-1$）互素的指数 e。要产生一个可用的密钥，必须找到两个大素数。这可以在计算机上借助本节前面提到的概率素性检测迅速完成。它们的乘积 $n = pq$，大约是 400 位的整数，不可能在短时间内被分解为因式。我们将看到，这正是没有一个独立的解密密钥就不可能迅速解密的重要原因。

2.5.8　RSA 加密

用 RSA 加密法时，信息被翻译成若干整数序列。为此可以先将每个字母翻译成整数，例如用凯撒密码翻译。这些整数再分成组，各组成为一个大整数，以代表一个字母段。加密过程是先把表示普通文字（即原信息）的整数 M 转换为表示密码文字（即加密信息）的整数 C，C 的计算公式是

$$C = M^e \bmod n$$

（为完成加密，可以使用快速指数取模的算法，例如本章末补充练习中第 14 题描述的算法。）加密后的信息以一段段数字的形式发送给预期的接受者。

例 10 说明了 RSA 怎样完成加密。为方便实际操作，例 10 中选用小素数 p 和 q，而不是 100 位或更大的素数。例中描述的密码尽管并不安全，但可以解释 RSA 密码使用的技术。

例 10　用 RSA 密码系统为信息 STOP 加密，其中 $p=43$、$q=59$，所以 $n=43\cdot 59=2537$。此外 $e=13$。注意

$$\gcd(e,(p-1)(q-1)) = \gcd(13, 42\cdot 58) = 1$$

解　我们把 STOP 的字母翻译成相应的等价数码，然后按 4 个数字一组分段。这样得到

1819　　1415

我们用映射

$$C = M^{13} \bmod 2537$$

为每一段加密。快速取模乘法计算得 $1819^{13} \bmod 2537 = 2081$ 及 $1415^{13} \bmod 2537 = 2182$。加密后的信息为 2081 2182。 ■

2.5.9　RSA 解密

如果知道解密密钥 d，即 e 模 $(p-1)(q-1)=1$ 的逆数，就可以很快恢复原信息。（由于 $\gcd(e,(p-1)(q-1))=1$，这一逆数一定存在。）要看出这一点，注意若 $de\equiv 1\ (\bmod (p-1)(q-1))$，则有整数 k，使 $de=1+k(p-1)(q-1)$。由此知

$$C^d = (M^e)^d = M^{de} = M^{1+k(p-1)(q-1)}$$

根据费马小定理（假定 $\gcd(M,p)=\gcd(M,q)=1$，这一关系不只在很少情况下成立），$M^{p-1}\equiv 1 (\bmod\ p)$ 及 $M^{q-1}\equiv 1(\bmod\ q)$。于是

$$C^d \equiv M\cdot(M^{p-1})^{k(q-1)} \equiv M\cdot 1 \equiv M (\bmod\ p)$$

及

$$C^d \equiv M\cdot(M^{q-1})^{k(p-1)} \equiv M\cdot 1 \equiv M (\bmod\ q)$$

由于 $\gcd(p,q)=1$，从中国余数定理知

$$C^d \equiv M (\bmod\ pq)$$

例 11 说明了怎样用 RSA 密码系统为信息解密。

例 11　我们收到的加密信息是 0981 0461。如果这是用例 10 中的 RSA 密码加密的，解密后的原信息是什么？

解　该信息是用 RSA 密码系统 $n=43.59$ 和指数 13 加密的。练习题 4 证明 $d=937$ 是 13 模 $42.58=2436$ 的逆数。我们用 937 作为解密指数。于是要为数字段 C 解密，计算

$$p = C^{937} \bmod 2537$$

为解密上述信息，用快速指数取模算法计算 0981^{937} mod 2537 = 0704 及 0461^{937} mod 2537 = 1115。从而原信息的数码形式是 0704 1115。翻译成英文字母，原信息为 HELP。 ■

2.5.10 用 *RSA* 作公钥系统

为什么 RSA 密码系统适合作公钥密码呢？如果我们知道模数 n 的因式分解，即知道素数 p 和 q，就可以用欧几里德算法迅速找到 e 模 $(p-1)(q-1)$ 的逆数 d。这使我们可以解密用我们的密钥发送的信息。迄今还不知道有不依赖 n 的因式分解的解密方法，或者说有任何也不导至 n 的因式分解的解密方法。已知最有效的分解法（到 1999 年止）需要数亿年才能分解 400 位的整数。于是若 p 和 q 都是 200 位的素数，用 $n = pq$ 作模数加密的信息除非已知素数 p 和 q，否则不可能在可以接受的时间内解密。

人们正在积极研究以求发现有效分解整数的新方法。几年以前还认为由于太大而不可能在合理的时间内分解的整数，现在已经可以程式化地分解了。超过 100 位的整数，以及若干超过 150 位的整数，已经被团队式努力而分解。一旦新的分解方法问世，就必须使用更大的素数以确保信息安全。不幸的是，先前认为安全的信息可能被并非接受方的人保存起来，然后当分解 RSA 加密使用的密钥中的 $n = pq$ 变得容易时解密。

RSA 方法已被实现，并用于某些特别敏感的应用中。不过最通用的密码系统仍是称为 DES（Data Encryption Standard，数据加密标准）的私钥系统。使用 DES 时，加密和解密均可在计算机上迅速完成。尽管有人相信用 DES 加密的信息可能被专家破译，大多数情况下 DES 仍被认为是足够安全的。有越来越多的人使用借助 RSA 系统的公钥密码系统，但 RSA 的加密和解密（借助最新一代计算机）对许多应用来说都太慢。不过有些应用既使用私钥又使用公钥。例如像 RSA 这样的公钥系统可以来为希望通信的双方分配私钥。然后这些人就可以用像 DES 这样的私钥系统来为信息加密和解密。

练习

1. 把下列各对整数的最大公约数表示为它们的线性组合。
   a) 10，11　　b) 21，44　　c) 36，48
   d) 34，55　　e) 117，213　　f) 0，223
   g) 123，2347　　h) 3454，4666　　i) 9999，11111
2. 把下列各对整数的最大公约数表示为它们的线性组合。
   a) 9，11　　b) 33，44　　c) 35，78
   d) 21，55　　e) 101，203　　f) 124，323
   g) 2002，2339　　h) 3457，4669　　i) 10001，13422
3. 证明 15 是 7 模 26 的逆数。
4. 证明 937 是 13 模 2436 的逆数。
5. 求 4 模 9 的逆数。
6. 求 2 模 17 的逆数。
7. 求 19 模 141 的逆数。
8. 求 144 模 233 的逆数。

*9. 若 a 和 m 是互素的正整数，证明 a 模 m 的逆是模 m 唯一的。[提示：假定同余式 $ax \equiv$

$1 \pmod{m}$有两个解 b 和 c。再用定理 2 证明$b \equiv c \pmod{m}$。]

10. 求证：若$\gcd(a, m) > 1$，则不存在 a 模 m 的逆。

11. 解同余方程$4x \equiv 5 \pmod 9$

12. 解同余方程$2x \equiv 7 \pmod{17}$。

*13. 若 m 是大于 1 的正整数，$ac \equiv bc \pmod m$，求证 $a \equiv b \pmod{m/\gcd(c, m)}$。

14. a) 求证小于 11 的正整数中除 1 和 10 以外的那些可以分成对，每一对中的两个整数互为模 11 的逆。

 b) 用 a) 中的结果证明 $10! \equiv -1 \pmod{11}$。

15. 求证若 p 为素数，则 $x^2 \equiv 1 \pmod p$的仅有的解是满足 $x \equiv 1 \pmod p$和 $x \equiv -1 \pmod p$的整数 x。

*16. a) 推广练习 14 中 a) 部分的结果，即证明若 p 为素数，则小于 p 的整数中除 1 和 $p-1$ 以外的那些可以分成对，每一对中的两个整数互为模 p 的逆。[提示：利用练习 15 中的结果。]

 b) 从 a) 可以断定，只要 p 是素数，则 $(p-1)! \equiv 1 \pmod p$。这一结果称为威尔逊 (Wilson) 定理。

 c) 若 n 为正整数，从 $(n-1)! \not\equiv 1 \pmod n$ 可以得什么结论?

*17. 本练习给出了费马小定理的一个证明轮廓。

 a) 假定 a 不能被素数 P 整除。证明整数 $1 \cdot a$，$2 \cdot a$，…，$(p-1)a$ 中任何两个都不是模 p 同余的。

 b) 从 a) 部分可以断定 1，2，…，$(p-1)$ 的乘积和 a，$2a$，…，$(p-1)a$ 的乘积是模 p 同余的。利用这一结论证明

$$(p-1)! \equiv a^{p-1}(p-1)! \pmod p$$

 c) 利用威尔逊定性（练习 16 中求证）证明只要 $p \nmid a$，$a^{p-1} \equiv 1 \pmod p$。

 d) 利用 c) 的结论证明 $a^p \equiv a \pmod p$对所有整数 a 成立。

18. 用中国余数定理证明任何使 $0 \leqslant a < m = m_1 m_2 \cdots m_n$ 的整数 a 都能唯一地表示为 n 元组 $(a \bmod m_1, a \bmod m_2, \cdots, a \bmod m_n)$，其中 m_1，m_2，…，m_n 是两两互素的整数。

*19. 令 m_1，m_2，…，m_n 为大于或等于 2 的整数且两两互素，求证如果 $a \equiv b \pmod{m_i}$，$i = 1, 2, \cdots, n$，则 $a \equiv b \pmod m$，其中 $m = m_1 m_2 \cdots m_n$。

*20. 证明模两两互素的一组整数的线性同余方程的共同解与这些模数之乘积是模唯一的，并以此完成中国余数定理的证明。[提示：假定 x 和 y 是两个共同解，证明对所有 i，$m_i \mid x-y$。再用练习 19 得 $m = m_1 m_2 \cdots m_n \mid x-y$。]

21. 哪些整数被 2 除余 1，被 3 除也余 1?

22. 哪些整数被 5 整除而被 3 除时余 1?

23. a) 用费马小定理证明 $2^{340} \equiv 1 \pmod{11}$，注意 $2^{340} = (2^{10})^{34}$。

 b) 利用 $2^{340} = (2^5)^{68} = 32^{68}$这一事实，证明 $2^{340} \equiv 1 \pmod{31}$。

 c) 从 a) 和 b) 推出结论 $2^{340} \equiv 1 \pmod{341}$。

24. a) 用费马小定理计算 $3^{302} \bmod 5$，$3^{302} \bmod 7$ 和 $3^{302} \bmod 11$。

 b) 利用 a) 中结果及中国余数定理求 $3^{302} \bmod 385$。(注意 $385 = 5 \cdot 7 \cdot 11$。)

25. a) 用费马小定理计算 $5^{2003} \bmod 7$，$5^{2003} \bmod 11$ 及 $5^{2003} \bmod 13$。
    b) 用 a) 中结果及中国余数定理求 $5^{2003} \bmod 1001$。(注意 $1001 = 7 \cdot 11 \cdot 13$。)
26. 求下列各对整数表示的小于 28 的非负整数 a，其中每对整数都表示$(a \bmod 4, a \bmod 7)$。
    a) (0, 0)　b) (1, 0)　c) (1, 1)
    d) (2, 1)　e) (2, 2)　f) (0, 3)
    g) (2, 0)　h) (3, 5)　i) (3, 6)
27. 用数对 ($a \bmod 3$, $a \bmod 5$) 表示小于 15 的每一个非负整数。
28. 解释怎样用练习 27 中求出的数对计算 4 加 7。
29. 求解例 8 中得到的同余系统。
*30. 证明如果 a 和 b 为正整数，则$(2^a - 1) \bmod (2^b - 1) = 2^{a \bmod b} - 1$
**31. 用练习 30 证明，若 a 和 b 为正整数，那么$\gcd(2^a - 1, 2^b - 1) = 2^{\gcd(a,b)} - 1$。[提示：证明用欧几里德算法计算$\gcd(2^a - 1, 2^b - 1)$时得到的余数是形为 $2^r - 1$ 的数，其中 r 是用欧几里德算法求$\gcd(a, b)$时产生的一个余数。]
32. 用练习 31 证明整数 $2^{35} - 1$，$2^{34} - 1$，$2^{33} - 1$，$2^{31} - 1$，$2^{29} - 1$ 和 $2^{23} - 1$ 是两两互素的。
33. 证明若 p 为奇素数，则麦逊尼数 $2^p - 1$ 的每个因数都是形为 $2kp + 1$ 的数，其中 k 是个非负整数。[提示：利用费马小定理和练习 31。]
34. 用练习 33 判断 $M_{13} = 2^{13} - 1 = 8191$ 和 $M_{23} = 2^{23} - 1 = 8\ 388\ 607$ 为素数。
*35. 证明如果知道 n 是两个素数 p 和 q 之积，且知道$(p-1) \cdot (q-1)$之值，则很容易分解 n 的因式。
36. 用 RSA 系统为信息 ATTACK 加密，其中 $n = 43 \cdot 59$，$e = 13$。把每个字母译成整数，再像例 10 那样组合整数对。
37. 如果用 RSA 系统加密以后的信息是 0667 1947 0671，其中 RSA 的参数是 $n = 43 \cdot 59$，$e = 13$，原信息是什么？(注：需使用计算工具才能在合理的时间内完成计算。)

如果 m 是正整数，整数 a 称为 m 的二次余数的条件是$\gcd(a, m) = 1$ 且同余式$x^2 \equiv a \pmod{m}$有解。换言之，m 的二次余数是与 m 互素的整数，而且是模 m 的完全平方。例如，2 是 7 的二次余数，因为$\gcd(2, 7) = 1$，且$3^2 \equiv 2 \pmod{7}$。3 不是 7 的二次余数，因为$\gcd(3, 7) = 1$，但$x^2 \equiv 3 \pmod{7}$无解。

38. 哪些整数是 11 的二次余数？
39. 求证若 p 是奇素数，a 是不能被 p 整除的整数，则同余式 $x^2 \equiv a \pmod{p}$要么没有解，要么恰有两个模 p 不同余的解。
40. 求证若 p 是奇素数，则在 1，2，…，$p-1$ 中恰有$(p-1)/2$ 个 p 的二次余数。若 p 是奇素数，a 是不能被 p 整除的整数，则勒让德符号 $\left(\frac{a}{p}\right)$ 定义为：在 a 为 p 的二次余数时为 1，否则为 -1。
41. 求证若 p 为奇素数，a 和 b 为整数，且 $a \equiv b \pmod{p}$，则

$$\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$$

42. 求证若 p 是奇素数且 a 是不能被 p 整除的正整数，则

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod p$$

43. 用练习 42 的结论证明若 p 是奇素数且 a 和 b 为不能被 p 整除的整数，则

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$$

44. 证明若 p 是奇素数，则只要 $p \equiv 1 \pmod 4$，-1 就是 p 的二次余数．若 $p \equiv 3 \pmod 4$，1 就不是 p 的二次余数。[提示：用练习 42 的结果。]
45. 求同余式 $x^2 \equiv 29 \pmod{35}$ 的所有解。[提示：求这一同余模 5 和模 7 的解，再利用中国余数定理。]
46. 求同余式 $x^2 \equiv 16 \pmod{105}$ 的所有解。[提示：求这一同余模 3、模 5 和模 7 的解，再利用中国余数定理。]

2.6 矩阵

2.6.1 引言

离散数学中用矩阵表示集合中元素之间的关系。在随后的章节中矩阵将用于各种不同的模型中。例如矩阵将用于通信网络模型和交通运输系统模型。许多算法都是用矩阵模型开发的。本节重温这些算法使用的矩阵运算。

定义 1 矩阵是数的矩形数组。m 行 n 列的矩阵称为 $m \times n$ 阶矩阵或 $m \times n$ 矩阵。行数和列数相同的矩阵称为方阵。若两个矩阵有同样多的行和列，而且每个位置上的对应项都相等，则这两个矩阵相等。

例 1

矩阵 $\begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$ 是 3×2 矩阵。 ■

现在介绍几个矩阵术语。黑体大写字母用来表示矩阵。

定义 2 令

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

A 的第 i 行是 $1 \times n$ 矩阵 $[a_{i1}\ a_{i2} \cdots a_{in}]$。**A** 的第 j 列是 $n \times 1$ 矩阵

$$\begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{bmatrix}$$

A 的第(i,j)项或第(i,j)元素是 a_{ij}，即第 i 行第 j 列上的数。方便的表示矩阵 A 的简写符号是$\mathbf{A}=[a_{ij}]$，表示 A 是第(i,j)元素为 a_{ij}的矩阵。

2.6.2 矩阵运算

我们从矩阵加法开始介绍矩阵的基本运算。

定义 3　令$\mathbf{A}=[a_{ij}]$和$\mathbf{B}=[b_{ij}]$为 $m\times n$ 矩阵。A 和 B 的和用 $\mathbf{A}+\mathbf{B}$ 表示，这是以 $a_{ij}+b_{ij}$为其第(i,j)元素的矩阵。换言之，$\mathbf{A}+\mathbf{B}=[a_{ij}+b_{ij}]$。

大小相同的两个矩阵的和是将它们对应位置上的元素相加得到的。大小不同的矩阵不能相加，因为两个矩阵的和只对行数和列数都一样的两个矩阵才有定义。

例 2　我们有

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & -3 \\ 3 & 4 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 & -1 \\ 1 & -3 & 0 \\ -1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -2 \\ 3 & -1 & -3 \\ 2 & 5 & 2 \end{bmatrix}$$

■

现在讨论矩阵乘积。只有在第一个矩阵的列数和第二个矩阵的行数相等时才能定义它们的乘积。

定义 4　令 A 为 $m\times k$ 矩阵，B 为 $k\times n$ 矩阵。A 和 B 的乘积 AB 是个 $m\times n$ 矩阵，其第(i,j)元素等于 A 的第 i 行和 B 的第 j 列对应元素乘积之和。换言之，若$\mathbf{AB}=[c_{ij}]$，则

$$c_{ij}=a_{i1}b_{1j}+a_{i2}b_{2j}+\cdots+a_{ik}b_{kj}=\sum_{t=1}^{k}a_{it}b_{tj}$$

在图 2-4 中，A 的彩色行和 B 的彩色列用于计算 AB 的元素 c_{ij}。当第一个矩阵的列数和第二个矩阵的行数不相等时，不能定义它们的乘积。

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ik} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2n} \\ \vdots & & & \vdots & & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kj} & \cdots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & c_{ij} & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$$

图 2-4　$\mathbf{A}=[a_{ij}]$和$\mathbf{B}=[b_{ij}]$的乘积

现在举几个矩阵乘积的例子。

例 3　令

$$\mathbf{A}=\begin{bmatrix}1&0&4\\2&1&1\\3&1&0\\0&2&2\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}2&4\\1&1\\3&0\end{bmatrix}$$

若 **AB** 有定义求 **AB**。

解 因为 **A** 是 4×3 矩阵而 **B** 是 3×2 矩阵，**A** 和 **B** 的乘积有定义，是 4×2 矩阵。要计算 **AB** 的元素，首先把 **A** 的行和 **B** 的列的对应元素相乘，然后再把这些乘积加起来。例如，**AB** 的（3，1）位置的元素是 **A** 的第 3 行和 **B** 的第 1 列对应元素的乘积之和，即 $3\cdot2+1\cdot1+0.3=7$。计算出 **AB** 的所有元素后，我们得

$$\mathbf{AB}=\begin{bmatrix}14&4\\8&9\\7&13\\8&2\end{bmatrix}$$ ■

矩阵乘法是不可交换的。也就是说，若 **A** 和 **B** 为矩阵，**AB** 和 **BA** 不一定相同。事实上可能这两个乘积中只有一个有定义。例如若 **A** 是 2×3 矩阵，**B** 是 3×4 矩阵，那么 **AB** 有定义，是 2×4 矩阵；但是 **BA** 没有定义，因为 3×4 矩阵和 2×3 矩阵无法相乘。

一般来说，假定 **A** 是 $m\times n$ 矩阵，**B** 是 $r\times s$ 矩阵，则只有在 $n=r$ 时 **AB** 有定义，$s=m$ 时 **BA** 有定义。不仅如此，既使 **AB** 和 **BA** 均有定义，也只有在 $m=n=r=s$ 时 **AB** 和 **BA** 才有相同的大小，即 **AB** 和 **BA** 必定均为同样大小的方阵。再进一步说，即使 **A** 和 **B** 均为 $n\times n$ 矩阵，**AB** 和 **BA** 也不一定相等。下面就是这样的一个例子。

例 4 令

$$\mathbf{A}=\begin{bmatrix}1&1\\2&1\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}2&1\\1&1\end{bmatrix}$$

是否 **AB** = **BA**？

解 我们得到

$$\mathbf{AB}=\begin{bmatrix}3&2\\5&3\end{bmatrix},\quad \mathbf{BA}=\begin{bmatrix}4&3\\3&2\end{bmatrix}$$

所以 **AB**≠**BA**。 ■

2.6.3 矩阵乘法运算

矩阵乘积的定义直接给出了计算两个矩阵乘积的算法。假定 $m\times n$ 矩阵 $\mathbf{C}=[c_{ij}]$ 是 $m\times k$ 矩阵 $\mathbf{A}=[a_{ij}]$ 和 $k\times n$ 矩阵 $\mathbf{B}=[b_{ij}]$ 的乘积。算法 1 中是用伪码表示的按矩阵乘积定义得到的算法。

我们可以用算法中使用的加法和乘法的次数判断这一算法的复杂性。

算法 1 矩阵乘法

```
procedure matrix multiplication (A,B : 矩阵)
for i := 1 to m
begin
    for j := 1 to n
    begin
      c_ij := 0
      for q := 1 to k
        c_ij := c_ij + a_iq b_qj
    end
end {C = [c_ij]是 A 和 B 的乘积}
```

例 5 用算法 1 计算两个 $n\times n$ 整数矩阵的乘积，需要做多少次整数加法和整数乘法？

解 在 A 和 B 的乘积中有 n^2 个元素，计算每个元素要做 n 次乘法和 $n-1$ 次加法。所以一共需要 n^3 次乘法和 $n^2(n-1)$ 次加法。■

令人吃惊的是，有比算法 1 效率高的矩阵乘法算法。例 5 说明直接定义计算两个 $n\times n$ 矩阵的乘积需要 $O(n^3)$ 次乘法和加法。用别的算法计算两个 $n\times n$ 矩阵的乘积只需 $O(n^{\sqrt{7}})$ 次乘法和加法。(本节末尾给出的推荐读物中可找到含这种算法细节的参考文献。)

涉及矩阵乘法的还有另一个重要问题，这就是怎样计算才能用最少的整数乘法计算乘积 $A_1A_2\cdots A_n$，其中 A_1，A_2，…，A_n 分别为 $m_1\times m_2$，$m_2\times m_3$，…，$m_n\times m_{n+1}$ 整数矩阵。(正如本节末尾例 13 所示，矩阵乘法是可结合的，所以，矩阵相乘的次序不影响计算结果。)研究这一问题之前，先要注意用算法 1 计算一个 $m_1\times m_2$ 矩阵和一个 $m_2\times m_3$ 矩阵需要做 $m_1m_2m_3$ 次乘法（参看本节末练习 23)。下面的例子解释了这一复杂性问题。

例 6 A_1，A_2 和 A_3 分别是 30×20，20×40 及 40×10 的整数矩阵，以什么次序计算 A_1，A_2 和 A_3 的乘积，使用的整数乘法次数最少？

解 有两种计算 $A_1A_2A_3$ 的次序，即 $A_1(A_2A_3)$ 和 $(A_1A_2)A_3$。若 A_2 和 A_3 首先相乘，须做 $20\cdot 40\cdot 10=8\ 000$ 次整数乘法来计算 20×10 矩阵 A_2A_3。然后需要 $30\cdot 20\cdot 10=6\ 000$ 次乘法来计算 A_1 和 A_2A_2 的乘积。因此共使用了 $8\ 000+6\ 000=14\ 000$ 次乘法。另一方面，若 A_1 和 A_2 首先相乘，须做 $30\cdot 40\cdot 20=24\ 000$ 次乘法来计算 30×40 矩阵 A_1A_2。然后需要 $30\cdot 40\cdot 10=12\ 000$ 次乘法以计算 A_1A_2 和 A_3 的乘积。因此共使用了 $24\ 000+12\ 000=36\ 000$ 次乘法。

显然，第一个计算顺序更有效。■

本书末推荐读物中讨论了判断 n 个矩阵相乘最有效方式的算法。

2.6.4 矩阵的转置和幂

现在引入一个以 0 和 1 为元素的重要矩阵。

定义 5 n 阶单位矩阵是 $n\times n$ 矩阵 $\mathbf{I}_n=[\delta_{ij}]$，其中 $\delta_{ij}=1$ 若 $i=j$，$\delta_{ij}=0$ 若 $i\neq j$。因此

$$\mathbf{I}_n=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}$$

用大小合适的单位矩阵乘一个矩阵不改变该矩阵。换言之，若 $\mathbf{A}$ 是 $m\times n$ 矩阵，则有

$$\mathbf{AI}_n=\mathbf{I}_m\mathbf{A}=\mathbf{A}$$

可以定义方阵的幂。若 $\mathbf{A}$ 是 $n\times n$ 矩阵，则有

$$\mathbf{A}^0=\mathbf{I}_n,\quad \mathbf{A}^r=\underbrace{\mathbf{AAA}\cdots\mathbf{A}}_{r\text{ 个}}$$

许多算法中都要使用把方阵的行和列交换的运算。

定义 6 令 $\mathbf{A}=[a_{ij}]$为 $m\times n$ 矩阵。$\mathbf{A}$ 的转置，用 $\mathbf{A}^t$ 表示，是交换 $\mathbf{A}$ 的行和列得到的 $n\times m$ 矩阵。换言之，若 $\mathbf{A}^t=[b_{ij}]$，则 $b_{ij}=a_{ji}$，$i=1,2,\cdots,n$，$j=1,2,\cdots,m$。

例 7 矩阵$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}$的转置是矩阵$\begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix}$。 ■

行列交换之后仍是原矩阵的那些矩阵往往有其重要性。

定义 7 若方阵 $\mathbf{A}$ 和它的转置相等，即 $\mathbf{A}=\mathbf{A}^t$，则 $\mathbf{A}$ 称为对称矩阵。因此$\mathbf{A}=[a_{ij}]$为对称矩阵的条件是对所有 i，j，$1\leqslant i\leqslant n$，$1\leqslant j\leqslant n$，$a_{ij}=a_{ji}$。

注意，矩阵对称的充分必要条件一是为方阵，二是对主对角线（由第 i 行第 i 列的元素组成，i 是行和列任一序数）对称。这一对称性如图 2-5 所示。

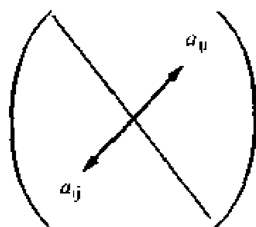


图 2-5 对称矩形

例 8 矩阵$\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}$是对称的。 ■

2.6.5 0－1 矩阵

元素非 0 即 1 的矩阵称为 0－1 矩阵。在第 6 章和第 7 章将会看到，0－1 矩阵常用来表

示离散结构。使用这些结构的算法的基础是以 0－1 矩阵做布尔运算。布尔运算基于由下式定义的一对字位上的运算 ∧ 和 ∨：

$$b_1 \wedge b_2 = \begin{cases} 1, & \text{若 } b_1 = b_2 = 1 \\ 0, & \text{否则} \end{cases}$$

$$b_1 \vee b_2 = \begin{cases} 1, & \text{若 } b_1 \text{ 或 } b_2 = 1 \\ 0, & \text{否则} \end{cases}$$

定义 8　令 $\mathbf{A}=[a_{ij}]$ 和 $\mathbf{B}=[b_{ij}]$ 为 $m \times n$ 阶 0－1 矩阵。**A** 和 **B** 的并，用 $\mathbf{A} \vee \mathbf{B}$ 表示，是个 0－1 矩阵，其 (i,j) 元素为 $a_{ij} \vee b_{ij}$。**A** 和 **B** 的交，用 $\mathbf{A} \wedge \mathbf{B}$ 表示，是个 0－1 矩阵，其 (i,j) 元素是 $a_{ij} \wedge b_{ij}$。

例 9　求 0－1 矩阵

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{和 } \mathbf{B} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

的并和交。

解　**A** 和 **B** 的并是

$$\mathbf{A} \vee \mathbf{B} = \begin{pmatrix} 1 \vee 0 & 0 \vee 1 & 1 \vee 0 \\ 0 \vee 1 & 1 \vee 1 & 0 \vee 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

A 和 **B** 的交是

$$\mathbf{A} \wedge \mathbf{B} = \begin{pmatrix} 1 \wedge 0 & 0 \wedge 1 & 1 \wedge 0 \\ 0 \wedge 1 & 1 \wedge 1 & 0 \wedge 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

■

现在我们定义两个矩阵的布尔乘积。

定义 9　令 $\mathbf{A}=[a_{ij}]$ 为 $m \times k$ 阶 0－1 矩阵，$\mathbf{B}=[b_{ij}]$ 为 $k \times n$ 阶 0－1 矩阵。**A** 和 **B** 的布尔乘积，用 $\mathbf{A} \odot \mathbf{B}$ 表示，是 $m \times n$ 矩阵 $[c_{ij}]$，其中

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdots \vee (a_{ik} \wedge b_{kj})$$

注意 **A** 和 **B** 的布尔乘积的计算方法类似于这两个矩阵的普通乘积，但要用运算 ∨ 代替加法，用运算 ∧ 代替乘法。我们给一个矩阵布尔乘法的例子。

例 10　求 **A** 和 **B** 的布尔乘积，其中

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

解　**A** 和 **B** 的布尔乘积 $\mathbf{A} \odot \mathbf{B}$ 由下式给出：

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) \\ (1 \wedge 1) \vee (0 \vee 0) & (1 \wedge 1) \vee (0 \vee 1) & (1 \wedge 0) \vee (0 \wedge 1) \end{bmatrix}$$

$$= \begin{bmatrix} 1\vee 0 & 1\vee 0 & 0\vee 0 \\ 0\vee 0 & 0\vee 1 & 0\vee 1 \\ 1\vee 0 & 1\vee 0 & 0\vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

■

算法 2 给出了计算两个矩阵的布尔乘积的伪码。

算法 2　布尔乘积

```
procedure Boolean product (A,B:0-1 矩阵)
for i := 1 to m
begin
    for j := 1 to n
    begin
       c_ij := 0
       for q := 1 to k
          c_ij := c_ij ∨ (a_iq ∧ b_qj)
    end
end {C = [c_ij]是 A 和 B 的布尔乘积}
```

我们还可以定义 0-1 方阵的布尔幂。这些幂将用于以后学习图中通路。通路常用于模拟类似计算机网络中的通信路径的对象。

定义 10　令 **A** 为 0-1 方阵，r 为正整数。**A** 的 r 次布尔幂是 r 个 **A** 的布尔乘积。**A** 的 r 次布尔幂用 $\mathbf{A}^{[r]}$ 表示，因此

$$\mathbf{A}^{[r]} = \underbrace{\mathbf{A}\odot\mathbf{A}\odot\mathbf{A}\odot\cdots\odot\mathbf{A}}_{r\text{ 个}}$$

（由于矩阵的布尔乘积是可结合的，这一定义是合理的。）我们常把 $\mathbf{A}^{[0]}$ 定义为 $\mathbf{I}_n$。

例 11　令 $\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$。对所有正整数 n 求 $\mathbf{A}^{[n]}$。

解　我们得到

$$\mathbf{A}^{[2]} = \mathbf{A}\odot\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

也得到

$$\mathbf{A}^{[3]} = \mathbf{A}^{[2]}\odot\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix},\quad \mathbf{A}^{[4]} = \mathbf{A}^{[3]}\odot\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

继续计算证明

$$\mathbf{A}^{[5]}=\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

读者现在可以发现 $\mathbf{A}^{[n]}=\mathbf{A}^{[5]}$对所有大于或等于 5 的正整数 n 均成立。 ■

很容易求出计算两个 $n\times n$ 矩阵的布尔乘积需要的字位运算次数。

例 12 若 **A** 和 **B** 为 $n\times n$ 阶 0－1 矩阵，计算 $\mathbf{A}\odot\mathbf{B}$ 需要做多少次字位运算？

解 $\mathbf{A}\odot\mathbf{B}$ 中有 n^2 个元素。用算法 2，需要 n 次 $\vee$ 和 n 次 $\wedge$ 来计算 $\mathbf{A}\odot\mathbf{B}$ 的一个元素。因此每求一个元素需要 $2n$ 次字位运算。所以用算法 2 计算 $\mathbf{A}\odot\mathbf{B}$ 需要 $2n^3$ 次字位运算。 ■

练习

1. 令 $\mathbf{A}=\begin{bmatrix}1&1&1&3\\2&0&4&6\\1&1&3&7\end{bmatrix}$。
   a) **A** 的大小阶是什么？
   b) **A** 的第 3 列是什么？
   c) **A** 的第 2 行是什么？
   d) **A** 在（3，2）位置上的元素是什么？
   e) $\mathbf{A}^t$ 是什么？
2. 求 **A**＋**B**，其中

   a) $\mathbf{A}=\begin{bmatrix}1&0&4\\-1&2&-2\\0&-2&-3\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}1&3&5\\2&2&-3\\2&-3&0\end{bmatrix}$

   b) $\mathbf{A}=\begin{bmatrix}-1&0&5&6\\-4&-3&5&-2\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}-3&9&-3&4\\0&-2&-1&2\end{bmatrix}$
3. 求 **AB**，若

   a) $\mathbf{A}=\begin{bmatrix}2&1\\3&2\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}0&4\\1&3\end{bmatrix}$

   b) $\mathbf{A}=\begin{bmatrix}1&-1\\0&1\\2&3\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}3&-2&-1\\1&0&2\end{bmatrix}$

   c) $\mathbf{A}=\begin{bmatrix}4&-3\\3&-1\\0&-2\\-1&5\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}-1&3&2&-3\\0&-1&4&-3\end{bmatrix}$

4. 求乘积 AB，其中

a) $\mathbf{A}=\begin{bmatrix}1 & 0 & 1\\ 0 & -1 & -1\\ -1 & 1 & 0\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}0 & 1 & -1\\ 0 & -1 & 0\\ -1 & 0 & 1\end{bmatrix}$

b) $\mathbf{A}=\begin{bmatrix}1 & -3 & 0\\ 1 & 2 & 2\\ 2 & 1 & -1\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}1 & -1 & 2 & 3\\ -1 & 0 & 3 & -1\\ -3 & -2 & 0 & 2\end{bmatrix}$

c) $\mathbf{A}=\begin{bmatrix}0 & -1\\ 7 & 2\\ -4 & -3\end{bmatrix}$，$\mathbf{B}=\begin{bmatrix}4 & -1 & 2 & 3 & 0\\ -2 & 0 & 3 & 4 & 1\end{bmatrix}$

5. 求矩阵 $\mathbf{A}$，使

$$\begin{bmatrix}2 & 3\\ 1 & 4\end{bmatrix}\mathbf{A}=\begin{bmatrix}3 & 0\\ 1 & 2\end{bmatrix}$$

［提示：需要解线性方程组来求 A。］

6. 求矩阵 $\mathbf{A}$，使

$$\begin{bmatrix}1 & 3 & 2\\ 2 & 1 & 1\\ 4 & 0 & 3\end{bmatrix}\mathbf{A}=\begin{bmatrix}7 & 1 & 3\\ 1 & 0 & 3\\ -1 & -3 & 7\end{bmatrix}$$

7. 令 $\mathbf{A}$ 为 $m\times n$ 矩阵，$\mathbf{0}$ 为元素全为 0 的 $m\times n$ 矩阵。证明 $\mathbf{A}=\mathbf{0}+\mathbf{A}=\mathbf{A}+\mathbf{0}$。

8. 证明矩阵加法是可交换的，即证明若 $\mathbf{A}$ 和 $\mathbf{B}$ 均为 $m\times n$ 矩阵，则 $\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$。

9. 证明矩阵加法是可结合的，即证明若 $\mathbf{A}$，$\mathbf{B}$ 和 $\mathbf{C}$ 均为 $m\times n$ 矩阵，则 $\mathbf{A}+(\mathbf{B}+\mathbf{C})=(\mathbf{A}+\mathbf{B})+\mathbf{C}$。

10. 令 $\mathbf{A}$ 为 3×4 矩阵，$\mathbf{B}$ 为 4×5 矩阵，$\mathbf{C}$ 是 4×4 矩阵。判断下列哪些乘积有定义，并求有定义的那些乘积的大小。

a) $\mathbf{AB}$　　b) $\mathbf{BA}$　　c) $\mathbf{AC}$

d) $\mathbf{CA}$　　e) $\mathbf{BC}$　　f) $\mathbf{CB}$

11. 如果乘积 $\mathbf{AB}$ 和 $\mathbf{BA}$ 均有定义，对矩阵 $\mathbf{A}$ 和 $\mathbf{B}$ 的大小我们了解多少？

12. 本练习要证明矩阵乘法对矩阵加法满足分配律。

a) 假定 $\mathbf{A}$，$\mathbf{B}$ 均为 $m\times k$ 矩阵，$\mathbf{C}$ 为 $k\times n$ 矩阵，求证 $(\mathbf{A}+\mathbf{B})\mathbf{C}=\mathbf{AC}+\mathbf{BC}$。

b) 假定 $\mathbf{C}$ 是 $m\times k$ 矩阵，$\mathbf{A}$ 和 $\mathbf{B}$ 为 $k\times n$ 矩阵，求证 $\mathbf{C}(\mathbf{A}+\mathbf{B})=\mathbf{CA}+\mathbf{CB}$。

13. 本练习要证明矩阵乘法满足结合律。假定 $\mathbf{A}$ 是 $m\times p$ 矩阵，$\mathbf{B}$ 是 $p\times k$ 矩阵，$\mathbf{C}$ 是 $k\times n$ 矩阵。求证 $\mathbf{A}(\mathbf{BC})=(\mathbf{AB})\mathbf{C}$。

14. $n\times n$ 矩阵 $\mathbf{A}=[a_{ij}]$ 称为对角矩阵，若对所有 $i\neq j$，$a_{ij}=0$。求证两个 $n\times n$ 对角矩阵的乘积仍是对角矩阵。给出一个计算这一乘积的简单规则。

15. 令 $\mathbf{A}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$。求计算 $\mathbf{A}^n$ 的公式，其中 n 为正整数。

16. 求证 $(\mathbf{A}^t)^t=\mathbf{A}$。

17. 令 $\mathbf{A}$ 和 $\mathbf{B}$ 为两个 $n\times n$ 矩阵。求证

a) $(\mathbf{A}+\mathbf{B})^t=\mathbf{A}^t+\mathbf{B}^t$　　b) $(\mathbf{AB})^t=\mathbf{B}^t\mathbf{A}^t$

若 **A** 和 **B** 是 $n\times n$ 矩阵，且 $\mathbf{AB}=\mathbf{BA}=\mathbf{I}_n$，则称 **B** 为 **A** 的逆矩阵（这一术语是合适的，因为这样的 **B** 是唯一的），而 **A** 是可逆的。符号 $\mathbf{B}=\mathbf{A}^{-1}$表示 **B** 是 **A** 的逆。

18. 证明 $\begin{bmatrix} 2 & 3 & -1 \\ 1 & 2 & 1 \\ -1 & 1 & 3 \end{bmatrix}$是矩阵$\begin{bmatrix} 7 & -8 & 5 \\ -4 & 5 & -3 \\ 1 & -1 & 1 \end{bmatrix}$的逆矩阵。

19. 令 **A** 为 2×2 矩阵，

$$\mathbf{A}=\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

证明若 $ab-bc\neq 0$，则

$$\mathbf{A}^{-1}=\begin{bmatrix} \dfrac{d}{ad-bc} & \dfrac{-b}{ad-bc} \\ \dfrac{-c}{ad-bc} & \dfrac{a}{ad-bc} \end{bmatrix}$$

20. 令

$$\mathbf{A}=\begin{bmatrix} -1 & 2 \\ 1 & 3 \end{bmatrix}$$

a) 求 $\mathbf{A}^{-1}$ [提示：用练习 19 的结果]。

b) 求 $\mathbf{A}^3$。

c) 求 $(\mathbf{A}^{-1})^3$。

d) 用 b) 和 c) 的答案证明 $(\mathbf{A}^{-1})^3=(\mathbf{A}^3)^{-1}$。

21. 令 **A** 为可逆矩阵。证明只要 n 是正整数，就有 $(\mathbf{A}^n)^{-1}=(\mathbf{A}^{-1})^n$。

22. 令 **A** 为矩阵。求证 $\mathbf{AA}^t$ 是对称的。[提示：借助练习 17b) 证明这一矩阵等于它的转置。]

23. 证明传统算法计算 $m_1\times m_2$ 矩阵 **A** 和 $m_2\times m_2$ 矩阵 **B** 的乘积需要 $m_1m_2m_3$ 次乘法。

24. 计算矩阵 $\mathbf{A}_1$，$\mathbf{A}_2$ 和 $\mathbf{A}_3$ 乘积的最有效方法是什么？$\mathbf{A}_1$，$\mathbf{A}_2$，$\mathbf{A}_3$ 的大小分别是

a) 20×50，50×10，10×40

b) 10×5，5×50，50×1

25. 计算 $\mathbf{A}_1$，$\mathbf{A}_2$，$\mathbf{A}_3$ 和 $\mathbf{A}_4$ 乘积的最有效方法是什么？其中 $\mathbf{A}_1$，$\mathbf{A}_2$，$\mathbf{A}_3$ 和 $\mathbf{A}_4$ 的大小分别是 10×2，2×5，5×20，20×3。

26. a) 求证以 x_1，x_2，…，x_n 为变量的线性方程组

$$\begin{aligned} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n&=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n&=b_2 \\ &\vdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n&=b_n \end{aligned}$$

可以表示为 $\mathbf{AX}=\mathbf{B}$，其中 $\mathbf{A}=[a_{ij}]$，**X** 是 $n\times 1$ 矩阵，且 x_i 就是它的第 i 行，**B** 是 $n\times 1$ 矩阵，b_i 是它的第 i 行。

b) 求证若 **A** 是可逆的（在练习 18 的前面定义了可逆），则 a) 中方程组的解可以用等式

$X = \mathbf{A}^{-1}\mathbf{B}$ 计算。

27. 用练习 18 和 26 解方程组

$$7x_1 - 8x_2 + 5x_3 = 5$$
$$-4x_1 + 5x_2 - 3x_3 = -3$$
$$x_1 - x_2 + x_3 = 0$$

28. 令 $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$，$\mathbf{B} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$。求

a) $\mathbf{A} \vee \mathbf{B}$　　b) $\mathbf{A} \wedge \mathbf{B}$　　c) $\mathbf{A} \odot \mathbf{B}$

29. 令 $\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$，$\mathbf{B} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$。求

a) $\mathbf{A} \vee \mathbf{B}$　　b) $\mathbf{A} \wedge \mathbf{B}$　　c) $\mathbf{A} \odot \mathbf{B}$

30. 求 $\mathbf{A}$ 和 $\mathbf{B}$ 的布尔乘积，其中

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

31. 令 $\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$。求

a) $\mathbf{A}^{[2]}$　　b) $\mathbf{A}^{[3]}$　　c) $\mathbf{A} \vee \mathbf{A}^{[2]} \vee \mathbf{A}^{[3]}$

32. 令 $\mathbf{A}$ 为 0－1 矩阵。证明

a) $\mathbf{A} \vee \mathbf{A} = \mathbf{A}$　　b) $\mathbf{A} \wedge \mathbf{A} = \mathbf{A}$

33. 本练习要证明交和并运算是可交换的。令 $\mathbf{A}$ 和 $\mathbf{B}$ 为 $m \times n$ 阶 0－1 矩阵。求证

a) $\mathbf{A} \vee \mathbf{B} = \mathbf{B} \vee \mathbf{A}$　　b) $\mathbf{A} \wedge \mathbf{B} = \mathbf{B} \wedge \mathbf{A}$

34. 本练习要证明交和并运算是可结合的。令 $\mathbf{A}$，$\mathbf{B}$ 和 $\mathbf{C}$ 为 $m \times n$ 阶 0－1 矩阵。证明

a) $(\mathbf{A} \vee \mathbf{B}) \vee \mathbf{C} = \mathbf{A} \vee (\mathbf{B} \vee \mathbf{C})$　　b) $(\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C} = \mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C})$

35. 本练习要建立交对并运算的分配律。令 $\mathbf{A}$，$\mathbf{B}$ 和 $\mathbf{C}$ 为 $m \times n$ 阶 0－1 矩阵。求证

a) $\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C})$　　b) $\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})$

36. 令 $\mathbf{A}$ 为 $n \times n$ 阶 0－1 矩阵。令 $\mathbf{I}$ 为 $n \times n$ 单位矩阵。证明 $\mathbf{A} \odot \mathbf{I} = \mathbf{I} \odot \mathbf{A} = \mathbf{A}$。

37. 本练习要证明 0－1 矩阵的布尔乘积是可结合的。假定 $\mathbf{A}$ 是 $n \times p$ 阶 0－1 矩阵，$\mathbf{B}$ 是 $p \times k$ 阶 0－1 矩阵，$\mathbf{C}$ 是 $k \times n$ 阶 0－1 矩阵。求证 $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$。

关键术语和结果

术语

算法：一组有限多条准确的指令，用于完成一个计算或解答一个问题

搜索算法：在表中定位元素的过程

线性搜索算法：逐个搜索表中元素的过程

对半搜索算法：逐次对半搜索有序表的过程

时间复杂性：算法解题需要的时间量

空间复杂性：算法解题需要的存储空间量

最坏情况时间复杂性：算法解答一定大小的问题需要的最大时间量

平均情况时间复杂性：算法解答一定大小的问题需要的平均时间量

$a \mid b$（a 能整除 b）：有整数 c 使 $b = ac$

素数：大于 1 且恰有两个正整数因子的正整数

合数：大于 1 又不是素数的正整数

麦逊尼素数：形为 $2^p - 1$ 的素数，其中 p 为素数

$\gcd(a,b)$（a 和 b 的最大公约数）：能整除 a 和 b 的最大整数

互素整数：使 $\gcd(a,b) = 1$ 的整数 a 和 b

两两互素的整数：一组整数其中任何两个整数都互素

$\text{lcm}(a,b)$（a 和 b 的最小公倍数）：能被 a 和 b 整除的最小正整数

$a \bmod b$：a 被正整数 b 除的余数

$a \equiv b \pmod m$（a 模 m 同余于 b）：$a - b$ 能被 m 整除

加密：使信息成为秘密的过程

解密：将加密信息还原的过程

$n = (a_k a_{k-1} \cdots a_1 a_0)_b$：$n$ 的 b 进制表示

二进制表示：整数以 2 为基数的表示

十六进制表示：整数以 16 为基数的表示

a 和 b 的整系数线性组合：形为 $sa + tb$ 的数，其中 s 和 t 为整数

a 模 m 的逆：使 $\bar{a}a \equiv 1 \pmod m$ 的整数 $\bar{a}$

线性同余：形为 $ax \equiv b \pmod m$ 的同余，其中 x 为变量

2 为基的伪素数：使 $2^{n-1} \equiv 1 \pmod n$ 的合数 n

私钥加密：加密密钥和解密密钥均须保密的加密法

公钥加密：加密密钥公开，解密密钥保密的加密法

矩阵：数的矩形数组

矩阵加法：见 2.6.2 节

矩阵乘法：见 2.6.2 节

$\mathbf{I}_n$（n 阶单位矩阵）：对角线元素为 1、其他元素为 0 的 $n \times n$ 阶矩阵

$\mathbf{A}^t$（$\mathbf{A}$ 的转置）：交换 $\mathbf{A}$ 的行和列得到的矩阵

对称：与自己的转置相等的矩阵是对称的

0－1 矩阵：元素非 0 即 1 的矩阵

$\mathbf{A} \vee \mathbf{B}$（$\mathbf{A}$ 和 $\mathbf{B}$ 的并）：见 2.6.5 节

$\mathbf{A} \wedge \mathbf{B}$（$\mathbf{A}$ 和 $\mathbf{B}$ 的交）：见 2.6.5 节

$\mathbf{A} \odot \mathbf{B}$（$\mathbf{A}$ 和 $\mathbf{B}$ 的布尔乘积）：见 2.6.5 节

结果

线性和对分搜索算法：（见 2.1 节）

算术基本定理：每个正整数均可唯一地表示为素数的乘积，其中素因子按从小到大的次序出现

除法算法：令 a 和 d 为整数，d 为正整数，则有唯一的整数 q 和 r，$0 \leqslant r < d$，使 $a = dq + r$

如果 a，b 为正整数，则 $ab = \gcd(a, b) \cdot \mathrm{lcm}(a, b)$。

欧几里德算法：用于求最大公约数（见2.4节算法1）

令 b 是大于1的正整数，则只要 n 是正整数，n 就能唯一地表示为 $n = a_k b^k + a_{k-1} b^{k-1} + \cdots a_1 b + a_0$ 的形式。

求整数的以 b 为基的展开式的算法（见2.4节算法2）。

传统的整数加法和乘法算法（见2.4节）。

两个整数的最大公约数可以表示为这两个整数的整系数线性组合。

如果 m 是正整数，$\gcd(a, m) = 1$，则 a 有唯一的模 m 逆。

中国余数定理：对一组两两互素的整数模同余的线性方程组有模这些模数之积的唯一解。

费马小定理：若 p 为素数且 $p \nmid a$，则 $a^{p-1} \equiv 1 \pmod{p}$

复习题

1. a) 定义术语算法。
   b) 有哪些描述算法的方式？
   c) 解题算法和解同样问题的计算机程序有何不同？
2. a) 用汉语描述一个求一列 n 个整数中最大、次大和第三大整数的算法。
   b) 用伪码表达这一算法。
   c) 这个算法需要做多少次比较？
3. a) 对一个求一串 n 个整数中最小整数的算法，给出最坏情况时间复杂性、平均情况时间复杂性和最好情况时间复杂性的定义（用它需要的比较次数）。
   b) 将一串 n 个整数中当前已找到的最小整数与每个整数比较以求其最小整数的算法，若用需要做的比较次数来度量，它的最坏情况、平均情况和最好情况时间复杂性是什么？
4. a) 描述一下在一串以增序排列的整数中找一个整数的线性搜索和对分搜索的算法。
   b) 比较一下上述两个算法的最坏情况时间复杂性。
   c) 这两个算法中是否一个总是比另一个快一些（用使用的比较次数度量）？
5. 陈述算术基本定理。
6. a) 描述一个将整数分解为素数因子的过程。
   b) 用这一过程分解80 707为素因子。
7. a) 定义两个整数的最大公约数。
   b) 给出至少三种求两个整数的最大公约数的方法，每个方法在什么情况下最有效？
   c) 求1 234 567和7 654 321的最大公约数。
   d) 求 $2^3 \cdot 3^5 \cdot 5^7 \cdot 7^9 \cdot 11$ 和 $2^9 \cdot 3^7 \cdot 5^5 \cdot 7^3 \cdot 13$ 的最大公约数。
8. a) 定义什么是 a 和 b 模7同余。
   b) -11，-8，-7，-1，0，3和17中有哪几对整数模7同余？
   c) 证明若 a 和 b 模7同余，则 $10a + 13$ 和 $-4b + 20$ 也模7同余。

9. 给出一个将整数的十进制表示转换为十六进制表示的过程。

10. a) 怎样求两个整数的（整系数）线性组合，使之等于它们的最大公约数？

 b) 将gcd(84,119)表达为 84 和 119 的线性组合。

11. a) $\overline{a}$ 为 a 模 m 的逆是什么意思？

 b) 若 m 是正整数且gcd$(a,m)=1$，怎样求 a 模 m 的逆？

 c) 求 7 模 19 的逆。

12. a) 在gcd$(a,m)=1$ 时，怎样用 a 模 m 的逆解线性同余方程 $ax\equiv b\ (\mathrm{mod}\ m)$？

 b) 解线性同余方程 $7x=13\ (\mathrm{mod}\ 19)$。

13. a) 陈述中国余数定理。

 b) 求同余方程组 $x\equiv1\ (\mathrm{mod}\ 4)$，$x\equiv2\ (\mathrm{mod}\ 5)$和 $x\equiv3\ (\mathrm{mod}\ 7)$的解。

14. 假定 $2^n-1\equiv1\ (\mathrm{mod}\ n)$。$n$ 必定是素数吗？

15. a) 什么是私钥密码系统和公钥密码系统的区别？

 b) 解释为什么移位密码是私钥系统。

 c) 解释为什么 RSA 密码系统是公钥系统？

16. 定义两个矩阵的乘积。何时这一乘积有定义？

17. a) 在矩阵乘积 $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$ 有定义时，有几种方法计算这一乘积，假定相继做两个矩阵的乘法。

 b) 假定 $\mathbf{A}_1$，$\mathbf{A}_2$，$\mathbf{A}_3$ 和 $\mathbf{A}_4$ 分别为 10×20，20×5，5×10 和 10×5 矩阵。怎样才能用最少次数的矩阵元素乘法来计算乘积 $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$？

补充练习

1. a) 给出一个算法为一列整数中最大整数的最后一次出现定位。

 b) 估计算法用了多少次比较。

2. a) 给出一个算法，求一列整数中最大和次大整数。

 b) 估计算法需用多少次比较？

3. a) 给出一个算法判断一个位串中是否含两个相邻的 0。

 b) 这一算法需用多少次比较？

4. a) 假定一列整数是按从大到小的次序排列的，而且每个整数都可以重复出现。设计一个算法为整数 x 在该列整数中的所有出现定位。

 b) 估计算法使用的比较次数。

5. 求与 5 模 17 同余的 4 个数。

6. 若 a 和 d 为正整数，求证有整数 q 和 r，使 $a=dq+r$，其中 $-d/2<r\leqslant d/2$

*7. 求证若 $ac\equiv bc\ (\mathrm{mod}\ m)$，则 $a\equiv b\ (\mathrm{mod}\ m/d)$，其中 $d=\gcd(m,c)$。

*8. 在 $100_{10}!$ 的二进制展开中尾部有多少个 0？

9. 用欧几里德算法求 10 233 和 33 341 的最大公约数。

10. 用欧几里德算法求 gcd（144，233）要做多少次除法？

11. 求gcd$(2n+1,3n+2)$，其中 n 是个正整数。[提示：用欧几里德算法。]

12. a) 假定 a，b 为正整数，$a\geqslant b$。求证：如果 $a=b$，则gcd$(a,b)=a$；如果 a 和 b 为偶数，则gcd$(a,b)=2\cdot$gcd$(a/2,b/2)$；如果 a 为偶数，b 是奇数，则gcd$(a,b)=$

gcd($a/2,b$)；如果 a 和 b 均为奇数，则gcd(a,b) = gcd($a-b,b$)。

b）利用 a）中结论，说明怎样不用除法，只用比较及二进制展开式的减法和移位来构造两个正整数的最大公约数。

c）用这一算法求gcd(1202,4848)。

13. 求证一个整数能被 9 整除的充分必要条件是它的十进数字之和能被 9 整除。

14. a）设计一个算法，用 n 的二进制表示计算 $x^n \bmod m$，其中 x 是整数，m 和 n 是正整数。[提示：连续求平方以计算 $x \bmod n$，$x^2 \bmod n$，$x^4 \bmod m$，等等。然后再用形为 $x^{2^k} \bmod m$ 的适当的幂相乘以求出 $x^n \bmod m$。]

b）估计这一算法使用的乘法次数。

一组整数称为是互素的，如果它们的最大公约数是 1。

15. 判断下列各组整数是否互素。

a）8，10，12　　b）12，15，25

c）15，21，28　　d）21，24，28，32

16. 找一组 4 个互素的整数，使它们之中任何两个都不互素。

17. a）假定用函数 $f(p) = (ap+b) \bmod 26$ 且 gcd(a，26) = 1 为信息加密。给出一个可用来解密的函数。

b）信息加密以后是 LJMKG MGMXF QEXMW。如果加密函数是 $f(p) = (7p+10) \bmod 26$，原信息是什么？

18. 求证同余方程组 $x \equiv 2 \pmod 6$ 和 $x \equiv 3 \pmod 9$ 无解。

19. 求出同余方程组 $x \equiv 4 \pmod 6$ 和 $x \equiv 13 \pmod{15}$ 的所有解。

*20. a）求证同余方程组 $x \equiv a_1 \pmod{m_1}$ 和 $x \equiv a_2 \pmod{m_2}$ 有解的充分必要条件是 $\gcd(m_1, m_2) \mid a_1 - a_2$

b）求证 a）中的解模 $\mathrm{lcm}(m_1, m_2)$ 是唯一的。

21. 若 $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$，求 $\mathbf{A}^n$。

22. 求证若 $\mathbf{A} = c\mathbf{I}$，其中 c 为实数，$\mathbf{I}$ 是 $n \times n$ 单位矩阵，则 $\mathbf{AB} = \mathbf{BA}$ 对所有 $n \times n$ 矩阵 $\mathbf{B}$ 成立。

23. 求证若 $\mathbf{A}$ 是 2×2 矩阵，且对任何 2×2 矩阵 $\mathbf{B}$，均有 $\mathbf{AB} = \mathbf{BA}$，则 $\mathbf{A} = c\mathbf{I}$，其中 c 是实数，$\mathbf{I}$ 是 2×2 单位矩阵。

$n \times n$ 矩阵称为上三角阵，如果只要 $i > j$，就有 $a_{ij} = 0$。

24. 从矩阵乘法的定义设计一个计算两个上三角阵乘积的算法，要求略去自然会为 0 的那些乘积。

25. 给出练习 24 中计算两个上三角阵乘积的算法的伪码描述。

26. 练习 25 中给出的算法使用多少次矩阵元素的乘法来计算两个 $n \times n$ 上三角矩阵的乘积？

27. 假定 $\mathbf{A}$ 和 $\mathbf{B}$ 均为可逆矩阵，且 $\mathbf{AB}$ 存在，求证 $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$。

28. 如果 $\mathbf{A}$，$\mathbf{B}$，$\mathbf{C}$，$\mathbf{D}$ 依次是 30×10，10×40，40×50 和 50×30 阶的矩阵，计算 $\mathbf{ABCD}$ 的最佳顺序是什么？假定计算 $p \times q$ 矩阵和 $q \times r$ 矩阵的乘积需要 pqr 次矩阵元素乘法。

29. 令 $\mathbf{A}$ 为 $n \times n$ 矩阵，再令 $\mathbf{0}$ 是元素全为 0 的 $n \times n$ 矩阵。证明下列各式为真。

a) $\mathbf{A}\odot\mathbf{0}=\mathbf{0}\odot\mathbf{A}=\mathbf{0}$ b) $\mathbf{A}\vee\mathbf{0}=\mathbf{0}\vee\mathbf{A}=\mathbf{A}$

c) $\mathbf{A}\wedge\mathbf{0}=\mathbf{0}\wedge\mathbf{A}=\mathbf{0}$

计算机题目

写程序以实现下列输入和相应输出。

1. 已知 n 个整数的表，找出其中最大整数。
2. 已知 n 个整数的表，找出最大整数在表中的首次和末次出现。
3. 已知 n 个不同整数的表，用线性搜索找出一个整数在表中位置。
4. 已知 n 个不同整数的有序表，用对分搜索找出一个整数在表中位置。
5. 已知 n 个不同整数的有序表和整数 x，找出用线性搜索和对分搜索确定 x 在表中位置需要做多少次比较。
6. 给定一个正整数，判断它是不是素数。
7. 用恺撒密码为给定的信息加密；已知用恺撒密码加密的一段信息，为此信息解密。
8. 用欧几里德算法求两个正整数的最大公约数。
9. 求两个正整数的最小公倍数。

*10. 求给定正整数的素因数分解。

11. 给定大于 1 的正整数 b，求正整数以 b 为基数的展开式。
12. 给定一个正整数，求它的康托展开式（参看 2.4 节练习 32 前的说明）。
13. 给定正整 n，模数 m，乘数 a，增量 c 和初值 x_0，它们满足 $0\leqslant a<m$，$0\leqslant c<m$，$0\leqslant x_0<m$，用线性同余产生函数 $x_{i+1}=(ax_i+c)\bmod m$ 生成一列 n 个伪随机数。
14. 已知正整数 a 和 b，求整数 s 和 t，使 $sa+tb=\gcd(a,b)$。
15. 给定 n 个模两两互素模的线性同余式，求模上述模数之积的这些同余式的共同解。
16. 已知 $m\times k$ 矩阵 $\mathbf{A}$ 和 $k\times n$ 矩阵 $\mathbf{B}$，求 $\mathbf{AB}$。
17. 已知方阵 $\mathbf{A}$ 和正整数 n，求 $\mathbf{A}^n$。
18. 判断给定方阵是否对称。
19. 已知 $n_1\times n_2$ 矩阵 $\mathbf{A}$，$n_2\times n_3$ 矩阵 $\mathbf{B}$，$n_3\times n_4$ 矩阵 $\mathbf{C}$ 和 $n_4\times n_5$ 矩阵 $\mathbf{D}$，它们的矩阵元素全为整数，找出计算这些矩阵乘积的最有效顺序（用计算使用的整数乘法和加法数量来度量。）
20. 给定两个 $m\times n$ 布尔矩阵，求它们的交和并。
21. 已知 $m\times k$ 布尔矩阵 $\mathbf{A}$ 和 $k\times n$ 布尔矩阵 $\mathbf{B}$，求 $\mathbf{A}$ 和 $\mathbf{B}$ 的布尔乘积 $\mathbf{AB}$。
22. 已知布尔方阵 $\mathbf{A}$ 和正整数 n，求 $\mathbf{A}^{[n]}$。

计算和研究

用计算程序或你自己写的程序完成下列练习。

1. 对不超过 100 的每个素数 p，判断 2^p-1 是否为素数。
2. 测试某个范围内的麦逊尼数 2^p-1 以判断它们是否为素数。（可以使用 GIMPS 课题提供的软件。）
3. 证明对 $0\leqslant n\leqslant 39$ 的所有整数 n，n^2+n+41 是素数，而当 $n=40$ 时不是素数。是否有 n

的整系数多项式，次数大于 0，当 n 为正整数时，它的值总是素数？

4. 求形如 n^2+1 的素数，其中 n 为正整数。尽可能多找出一些。现在还不知道这样的素数是否有无穷多个。
5. 找出 10 个不同的素数，每个都有 100 位数。
6. 小于 1 000 000 的素数有多少个？小于 10 000 000 的素数有多少个？小于 100 000 000 的呢？对于正整数 x，你能否提出对小于 x 的素数的个数的一个估计？
7. 求随机选取的 10 个 20 位的不同奇数的素数分解。记录分解每个整数消耗的时间。对 10 个 30 位的不同奇数，40 位的不同奇数等重复上述作业。尽可能多做。
8. 求基数为 2 的所有伪素数，即找出使 $2^{n-1} \equiv 1 \pmod{n}$ 的所有整数 n，其中 n 不超过 10 000。

写作题目

用课外资料就下列课题写出短文。

1. 查一查算法这一单词的历史，描述早期著作中这一单词的用法。
2. 描述并行算法的含义。说明本书使用的伪码怎样扩展才能处理并行算法。
3. 解释怎样度量并行算法的复杂性。给出例子以阐明这一概念，并说明并行算法可以比没有并行操作的算法更快完成任务。
4. 描述用于检查麦逊尼数是否为素数的鲁卡斯－莱莫（Lucas-Lehmer）测试。讨论一下 GIMPS 课题在用这一测试寻找麦逊尼素数方面的进展。
5. 解释实践中怎样用概率素性测试来产生特别大的几乎肯定是素数的数。这种测试是否有什么潜在的弊端？
6. 卡米克尔数（Carmichal number）指的是那些整数，以与其互素的整数为基数时它们都是伪素数。75 年以前提出的是否有无穷多个卡米克尔数的问题当今得到了解答。解释什么是卡米克尔数，给出几个这种数的例子，描述一下在证明这种数有无穷多个时涉及的要点。
7. 围绕分解素数算法的复杂性以及能分解的数的大小，总结一下它们的现状。你认为何时才有因式分解 200 位数的实际可能性？
8. 描述一下现代计算机中使用的正整数加、减、乘、除的算法。
9. 描述中国余数定理的历史。描述一下中国和印度著作中提出的相关问题以及怎样将中国余数定理用于这些问题。
10. 怎样得到一串真正的随机数，而不是伪随机数？已经查明用伪随机数做模拟或试验有什么缺点？伪随机数的那些性质是随机数不该有的？
11. 描述如何应用公钥密码。用于给定因式分解算法状态的安全是公钥密码的应用途径吗？用公钥密码保证安全的信息将来会成为不安全的吗？
12. 描述如何才能把公钥密码用于签字的秘密信息，使接收者确信信息是由声明发送的人所发出的。
13. 说明如何用同余确定任何给定的日期是星期几。
14. 描述某些能用于求大整数乘积的有效算法。
15. 描述某些能用于求大矩阵相乘的有效算法。

第3章 数学推理

为了理解书面数学，就必须理解正确的数学论证（即证明）是由什么组成的。为了学习数学，就需要构造数学论证而不仅仅是阅读说明。显然，这样做就要求理解建立证明所用到的技术。本章目的在于讲授正确的数学论证是由什么组成的，并且给予学生构造这些论证所必需的工具。

注意，马上就要学习的建立证明的方法也在计算机科学里到处使用，比如计算机推理所用的规则，验证程序是正确的用到的技术，以及使用自动推理来构造新的定理所用的规则等。

许多数学命题都断言对所有的正整数来说某个性质为真。这样命题的例子如：对每个正整数 n 来说，$n! \leqslant n^n$，n^3-n 被 3 除尽，前 n 个正整数之和是 $n(n+1)/2$ 等。本章和本书的主要目的之一就是让学生彻底理解证明这种类型的结果所用的数学归纳法。

在前几章里以明显的方式定义了集合、序列和函数。也就是说，通过列举其中的元素或给出某个刻划这些元素的性质来描述集合。对于序列的项和函数的值则给出公式。存在另一种重要的方式来定义这些对象，它以数学归纳法为基础。为了定义序列和函数，就规定某些初始的项，并且给出从已知的值求后续的值的规则。例如，可以这样定义序列 $\{2n\}$：规定 $a_1=2$ 和对 $n=1,2,3,\cdots$ 来说 $a_{n+1}=2a_n$。可以这样定义集合：列举其中一部分元素，并且给出从这些已知属于集合的元素来构造其他元素的规则。这样的定义在离散数学和计算机科学中到处使用，称为递归定义。

当指定解决一个问题的过程时，这个过程总是正确地解决这个问题。仅仅通过测试来查看对一组输入值获得了正确的结果，还不能说明这个过程总是正确的。只有证明了过程总是产生正确的结果，才能保证这个过程的正确性。本章最后一节包含对程序验证技术的介绍。这是一种验证过程为正确的形式化技术。程序验证是正在进行的以机械方式证明程序为正确的各种尝试的基础。

3.1 证明方法

3.1.1 引言

在数学研究中提出的两个重要问题是：（1）什么时候数学论证是正确的？（2）什么方法可以用来构造数学论证？本节通过描述各种形式的正确与不正确的数学论证来帮助回答这些问题。

定理是可以被证明为真的命题。用一系列命题来证明一条定理为真，这些命题就形成一项论证，称为证明。在证明里用到的命题可以包括：公理或公设（它们是关于数学结构的基本假设），被证明定理本身的前提，以及从前证明过的定理等。推理规则（它们是从其他断言得出结论所用的方法）把证明的各个步骤联系起来。

本节将讨论推理规则。这样做有助于解释清楚正确的证明是由什么组成的。还将描述某

些常见形式的不正确的推理（称为谬误）。然后将介绍各种常用的证明定理的方法。

注意　名词“引理”和“推论”用于特定类型的定理。**引理**是在其他定理的证明中所用的简单定理。（例如第 2.4 节引理 1，用它证明定理：欧几里德算法产生两个整数的最大公约数。）当使用了一系列的引理（其中每个引理都被单独地证明）时，一些复杂的证明通常会更容易被理解。**推论**是从已经证明了的定理直接证实的命题。“**猜想**”是真值未知的命题。当发现了猜想的证明时，这个猜想就成为定理。许多时候猜想都被证明为假，所以猜想不是定理。

本章讨论的证明方法是重要的，不仅因为它们被用来证明数学定理，而且因为它们对计算机科学的诸多应用。这些应用包括：验证计算机程序是正确的，证明操作系统是安全的，在人工智能领域里进行推理，等等。所以，理解证明中所用到的技术无论在数学里还是在计算机科学里都是必要的。

3.1.2　推理规则

现在将要介绍命题逻辑的推理规则。这些规则为用来证明一个结论是从一组前提合乎逻辑地得出的步骤提供了正当理由。重言式$(p\wedge(p\rightarrow q))\rightarrow q$ 是一个被称为*假言推理*或*分离规则*的推理规则的基础。这个重言式写成下列方式：

$$\begin{array}{l} p \\ \underline{p\rightarrow q} \\ \therefore q \end{array}$$

使用这种记号时，前提写成一列，而结论写在横线下。（符号∴表示“因此”。）假言推理是说，若蕴涵式及其前件都已知为真，则这个蕴涵式的后件为真。

例 1　假定蕴涵式“若今天下雪，则将去滑雪”和它的前件“今天正在下雪”都为真。那么，根据假言推理，得出蕴涵式的后件“将去滑雪”为真。■

例 2　蕴涵式“若 n 被 3 整除，则 n^2 被 9 整除”为真。所以，若 n 被 3 整除，则根据假言推理得出 n^2 被 9 整除。■

表 3-1　推理规则

推理规则	重言式	名称
p / $\therefore p\vee q$	$p\rightarrow(p\vee q)$	附加
$p\wedge q$ / $\therefore p$	$(p\wedge q)\rightarrow p$	化简
p, q / $\therefore p\wedge q$	$((p)\wedge(q))\rightarrow(p\wedge q)$	合取
p, $p\rightarrow q$ / $\therefore q$	$[p\wedge(p\rightarrow q)]\rightarrow q$	假言推理

（续）

推理规则	重言式	名称
$\neg q$ $p \to q$ $\therefore \neg p$	$[\neg q \wedge (p \to q)] \to \neg p$	取拒式
$p \to q$ $q \to r$ $\therefore p \to r$	$[(p \to q) \wedge (q \to r)] \to (p \to r)$	假言三段论
$p \vee q$ $\neg p$ $\therefore q$	$[(p \vee q) \wedge \neg p] \to q$	析取三段论

表 3-1 列出了重要的推理规则。在第 1.2 节的练习里可以找到对这些推理规则的验证。这里是使用这些推理规则的一些例子。

例 3 说出哪个推理规则是下列论证的基础："现在气温在冰点以下。因此，要么现在气温在冰点以下，要么现在下雨。"

解 设 p 是命题"现在气温在冰点以下"，而 q 是命题"现在正在下雨"。那么这个论证形如

$$\frac{p}{\therefore p \vee q}$$

这是使用附加规则的论证。■

例 4 说出哪个推理规则是下列论证的基础："现在气温在冰点以下并且现在下雨。因此，现在气温在冰点以下。"

解 设 p 是命题"现在气温在冰点以下"，而 q 是命题"现在正在下雨"。这个论证形如

$$\frac{p \wedge q}{\therefore p}$$

这个论证使用化简规则。■

例 5 说出在下列论证里使用哪个推理规则：

若今天下雨，则我们今天将不野餐。若我们今天不野餐，则我们明天将野餐。因此，若今天下雨，则我们明天将野餐。

解 设 p 是命题"今天下雨"，设 q 是命题"我们今天将不野餐"，而设 r 是命题"我们明天将野餐"。则这个论证形如

$$\frac{\begin{array}{l} p \to q \\ q \to r \end{array}}{\therefore p \to r}$$

因此，这个论证是假言三段论。■

若每当所有的前提都为真时，结论也为真，则这样的论证称为有效的。所以，证明从前提 p_1，p_2，…，p_n 合乎逻辑地得出 q，就等于证明蕴涵式

$$(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \rightarrow q$$

为真。当在有效论证里用到的所有命题都为真时，就得出正确的结论。不过，当在有效论证里用到一个或多个假命题时，该论证可能得出不正确的结论。例如，“若 101 被 3 整除，则 101^2 被 9 整除。101 被 3 整除。所以，101^2 被 9 整除。”是一个基于假言推理的有效推理。不过，这个推理的结论为假，因为 9 不能整除 $101^2 = 10201$。在这个论证里使用了假命题“101 被 3 整除”，这就意味着该论证的结论可能为假。

当存在许多前提时，为了证明一个论证是有效的，就常常需要多个推理规则。对此通过下面这些例子来说明。在这些例子中一步一步地显示出论证的步骤，明确地叙述出每步的理由。这些例子也说明如何使用推理规则来分析以用语言表述的论证。

例 6　证明：前提“今天下午没有出太阳并且今天比昨天冷”，“只有今天下午出太阳，我们才将去游泳”，“若我们不去游泳，则我们将乘独木舟游览”，以及“若我们乘独木舟游览，则我们将在黄昏时回家”，导致结论“我们将在黄昏时回家”。

解　设 p 是命题“今天下午出太阳”，q 是命题“今天比昨天冷”，r 是命题“我们将去游泳”，s 是命题“我们将乘独木舟游览”，而 t 是命题“我们将在黄昏时回家”。则这些前提成为 $\neg p \wedge q, r \rightarrow p, \neg r \rightarrow s, s \rightarrow t$。结论是 t。

如下构造一个论证来证明这些前提得出需要的结论。

步骤	理由
1. $\neg p \wedge q$	前提引入
2. $\neg p$	化简，用步骤 1
3. $r \rightarrow p$	前提引入
4. $\neg r$	取拒式，用步骤 2 和 3
5. $\neg \rightarrow s$	前提引入
6. s	假言推理，用步骤 4 和 5
7. $s \rightarrow t$	前提引入
8. t	假言推理，用步骤 6 和 7

■

例 7　证明：前提“若你发给我电子邮件消息，则我将完成编写程序”，“若你不发给我电子邮件消息，则我将早早地去睡觉”，以及“若我早早地去睡觉，则我将感觉精力充沛地醒来”，导致结论“若我不完成编写程序，则我将感觉精力充沛地醒来”。

解　设 p 是命题“你发给我电子邮件消息”，q 是命题“我将完成编写程序”，r 是命题“我将早早地去睡觉”，而 s 是命题“我将感觉精力充沛地醒来”。则这些前提是 $p \rightarrow q$，$\neg p \rightarrow r, r \rightarrow s$。需要的结论是 $\neg q \rightarrow s$。

下列论证证明这些前提得出需要的结论。

步骤	理由
1. $p \rightarrow q$	前提引入
2. $\neg q \rightarrow \neg p$	步骤 1 的逆否命题
3. $\neg p \rightarrow r$	前提引入
4. $\neg q \rightarrow r$	假言三段论，用步骤 2 和 3

5. $r \to s$　　前提引入
6. $\neg q \to s$　　假言三段论，用步骤4和5 ■

3.1.3 谬误

几种常见的谬误都来源于不正确的论证。这些谬误看上去像是推理规则，但是它们是基于偶然事件而不是重言式。在这里讨论这些谬误，是为了说明在正确与不正确的推理之间的区别。

命题$[(p \to q) \land q] \to p$不是重言式，因为当$p$为假而$q$为真时，它为假。不过，存在许多把它当作重言式的不正确论证。这种类型的不正确推理称为*肯定结论谬误*。

例8　下列论证是否有效？

若你做本书的每一道练习，则你将学习离散数学。你学习过离散数学。
因此，你做过本书的每一道练习。

解　设p是命题“你做过本书的每一道练习”。设q是命题“你学习过离散数学”。这个论证形如：若$p \to q$并且q，则p。这是使用肯定结论谬误的不正确推理的例子。事实上，你可能通过其他某种方式学习离散数学而没有做本书的每一道练习。（你可能通过阅读、听讲座、做本书的一些但不是全部练习等方式来学习离散数学。） ■

例9　设p是命题“$n \equiv 1 \pmod 3$”，设q是命题“$n^2 \equiv 1 \pmod 3$”。蕴涵式$p \to q$为真，它是“若$n \equiv 1 \pmod 3$，则$n^2 \equiv 1 \pmod 3$”。若q为真，即$n^2 \equiv 1 \pmod 3$为真，则是否得出p为真，即$n \equiv 1 \pmod 3$？

解　得出p为真这是不正确的，因为或许$n \equiv 2 \pmod 3$。若得出p为真的不正确结论，则这是肯定结论谬误的例子。 ■

命题$[(p \to q) \land \neg p] \to \neg q$不是重言式，因为当$p$为假而$q$为真时，它为假。许多不正确的论证都不正确地把它当作了推理规则。这种类型的不正确推理称为*否定假设谬误*。

例10　设p和q如同例8那样。若蕴涵式$p \to q$为真，并且$\neg p$为真，则得出$\neg q$为真这是否正确？换句话说，如果假定：若你做本书里每一道练习，则你将学习离散数学，那么假定：若你没有做过本书里每一道练习，则你没有学习离散数学，这是否正确？

解　即使你没有做过本书里每一道练习，你也可能学过离散数学。这个不正确的论证具有形式：$p \to q$和$\neg p$蕴涵$\neg q$，这是否定假设谬误的例子。 ■

例11　设p和q如同例9那样。利用$p \to q$为真的事实，假定若$\neg p$为真，则$\neg q$为真，这是否正确？换句话说，利用蕴涵式若$n \equiv 1 \pmod 3$，则$n^2 \equiv 1 \pmod 3$，得出若$n \not\equiv 1 \pmod 3$则$n^2 \not\equiv 1 \pmod 3$，这是否正确？

解　得出若$n \not\equiv 1 \pmod 3$，则$n^2 \not\equiv 1 \pmod 3$，这是不正确的，因为当$n \equiv 2 \pmod 3$时，$n^2 \not\equiv 1 \pmod 3$。这个不正确的论证是否定假设谬误的另一个例子。 ■

许多不正确的论证都是基于一种被称为回避正题的谬误。当证明中的一步或多步是基于被证明的命题为真时，就出现这种谬误。换句话说，当用命题自身或与其等价的命题去证明

该命题时就产生这种谬误。所以这种谬误也称为循环论证。

例 12　下列论证是否正确？它声称证明：每当 n^2 是偶数时，n 就是偶数。

假定 n^2 是偶数。则对某个整数 k 来说有 $n^2=2k$。设对某个整数 l 来说有 $n=2l$。这样就证明了 n 是偶数。

解　这个论证是不正确的。证明里出现命题"设对某个整数 l 来说有 $n=2l$。"没有给出任何论证来证明它为真。这是循环论证，因为这个命题等价于被证明的命题，即"n 是偶数"。当然，这个结果本身是正确的；只是这个证明方法是错误的。■

3.1.4　带量词命题的推理规则

已经讨论了命题的推理规则。现在将要描述包含量词的命题的一些重要的推理规则。在数学论证里大量地使用这些推理规则，通常都没有明确地指出来。

*全称量词消去*是这样的推理规则，用它从前提 $\forall xP(x)$ 得出 $P(c)$ 为真，其中 c 是论域里的具体成员。当从命题"所有女人都聪明"得出"丽沙聪明"时，就使用了全称量词消去，其中丽沙是论域所有女人中的一员。

*全称量词引入*是这样的推理规则，它说在对论域里所有元素 c 来说 $P(c)$ 都为真的前提下，$\forall xP(x)$ 为真。当通过从论域里拿出一个任意元素 c 并证明 $P(c)$ 为真来证明 $\forall xP(x)$ 为真时，就使用了全称量词引入。所选择的元素 c 必须是论域里一个任意的元素，而不是特定的元素。在许多数学证明里隐含地使用全称量词引入，而很少明确地指出来。

*存在量词消去*是这样的推理规则，它允许从已知 $\exists xP(x)$ 为真，得出在论域里存在一个使得 $P(c)$ 为真的元素 c。在这里不能选择一个任意值的 c，而必须是使得 $P(c)$ 为真的那个 c。通常不知道 c 是什么，仅仅知道它存在。因为它存在，所以可以给它一个名称（c）而继续论证。

*存在量词引入*是这样的推理规则，用它在已知使 $P(c)$ 为真的一个具体的 c 时，得出 $\exists xP(x)$ 为真。即若知道论域里一个使 $P(c)$ 为真的元素 c，则知道 $\exists xP(x)$ 为真。

这些推理规则总结在表 3-2 里。

表 3-2　带量词的命题的推理规则，U 是论域

推理规则	名　称
$\forall xP(x)$ ——————— $\therefore P(c)$，若 $c\in U$	全称量词消去
$P(c)$，对任意 $c\in U$ ——————— $\therefore \forall xP(x)$	全称量词引入
$\exists xP(x)$ ——————— $\therefore P(c)$，对某个元素 $c\in U$	存在量词消去
$P(c)$，对某个元素 $c\in U$ ——————— $\therefore \exists xP(x)$	存在量词引入

在下一个例子里将要说明如何使用带量词的命题的推理规则之中的一个。

例 13　证明前提"在本离散数学课堂的每一个人学过一门计算机课程"和"玛拉是本

课堂的学生”蕴涵结论“玛拉学过一门计算机课程”。

解 设$D(x)$表示“x 在本离散数学课堂”，并且设$C(x)$表示“x 学过一门计算机课程。”则前提是$\forall x(D(x)\rightarrow C(x))$和 D(玛拉)。结论是 C(玛拉)。

下列步骤可以用来从前提证明结论。

步骤	理由
1. $\forall x(D(x)\rightarrow C(x))$	前提引入
2. D(玛拉)$\rightarrow C$(玛拉)	全称量词消去，用步骤 1
3. D(玛拉)	前提引入
4. C(玛拉)	假言推理，用步骤 2 和 3

■

注意 数学论证常常包含既使用命题推理规则又使用量词推理规则的步骤。例如，全称量词消去和假言推理就常常一起使用。当把这些推理规则组合起来时，从前提$\forall x(P(x)\rightarrow Q(x))$和 $P(c)$就证明结论 $Q(c)$为真，其中 c 是论域的成员。

注意 数学里许多定理说：对一个具体集合（比如整数集或实数集）里的所有元素来说成立一个性质。虽然这些定理的准确陈述需要包含全称量词，但是数学里的标准约定是省略全称量词。例如，命题“若整数 n 被 3 整除，则 n^2 被 9 整除”其实意味着“对所有整数 n 来说，若整数 n 被 3 整除，则 n^2 被 9 整除”。同理，命题“若 $x>y$，其中 x 和 y 都是正实数，则 $x^2>y^2$”其实意味着“对所有正实数 x 和 y 来说，若 $x>y$，则 $x^2>y^2$”。另外，当证明这种类型的定理时，常常使用全称量词引入规则而不明确地指出来。证明的第一步通常涉及选择论域里的一个一般元素。随后的步骤证明这个元素具有所考虑的性质。全称量词引入蕴涵着对论域里所有元素来说定理都成立。

在随后的讨论里将遵循通常的约定而不明确地指出使用了全称量词和全称量词引入。不过，读者应当总是能够理解何时隐含地应用了这条推理规则。

3.1.5 证明定理的方法

在第 1 章和第 2 章里证明过几个定理。现在进一步明确一下构造证明的方法。将要描述如何证明不同类型的命题。

因为许多定理都是蕴涵式，所以证明蕴涵式的技术是重要的。回忆一下：$p\rightarrow q$ 为真，除非 p 为真并且 q 为假。注意当证明命题 $p\rightarrow q$ 时，只需要证明：若 p 为真则 q 为真；通常并不要证明 q 为真。下面的讨论将给出最常见的证明蕴涵式的技术。

可以通过证明：若 p 为真则 q 也必然为真，来证明蕴涵式 $p\rightarrow q$。这样就证明 p 为真而 q 为假的组合永远不会出现。这种证明称为*直接证明*。为了完成这样的证明，假定 p 为真，并且使用推理规则和已经证明的定理，来证明 q 也必然为真。

例 14 给出定理“若 n 是奇数，则 n^2 是奇数”的直接证明。

解 假定这个蕴涵式的前件为真。即假定 n 是奇数。则 $n=2k+1$，其中 k 是整数。由此得出$n^2=(2k+1)^2=4k^2+4k+1=2(2k^2+2k)+1$。因此，$n^2$ 是奇数（它是一个整数的两倍再加 1）。 ■

因为蕴涵式 $p\to q$ 等价于它的逆否命题 $\neg q\to\neg p$，所以可以通过证明它的逆否命题 $\neg q\to\neg p$ 为真，来证明蕴涵式 $p\to q$。通常直接地证明这个相关的蕴涵式，但是任何证明技术都可以使用。这种类型的论证称为*间接证明*。

例 15 给出定理“若 $3n+2$ 是奇数，则 n 是奇数”的间接证明。

解 假定这个蕴涵式的后件为假：即假定 n 是偶数。则对某个整数 k 来说有 $n=2k$。所以 $3n+2=3(2k)+2=6k+2=2(3k+1)$，所以 $3n+2$ 是偶数（因为它是 2 的倍数）。因为对这个蕴涵式后件的否定蕴涵着前件为假，所以原来的蕴涵式为真。 ■

假定蕴涵式 $p\to q$ 的前件 p 为假。则蕴涵式 $p\to q$ 为真，因为该命题形如 **F→T** 或 **F→F**，所以它为真。因此，若可以证明 p 为假，则可以给出蕴涵式 $p\to q$ 的证明，这称为*空证明*。常常用空证明来证明一些定理的特殊情形，这些定理说：对所有正整数来说，一个蕴涵式为真［即形如 $\forall nP(n)$ 的定理，其中 $P(n)$ 是命题函数］。这种定理的证明技术将在 3.2 节讨论。

例 16 证明命题 $P(0)$ 为真，其中 $P(n)$ 是命题函数“若 $n>1$，则 $n^2>n$”。

解 注意命题 $P(0)$ 是蕴涵式“若 $0>1$，则 $0^2>0$”。因为前提 $0>1$ 为假，所以蕴涵式 $P(0)$ 自动地为真。 ■

注意 *蕴涵式的后件 $0^2>0$ 为假这个事实，与蕴涵式的真值无关，因为前件为假的蕴涵式保证为真。*

假定蕴涵式 $p\to q$ 的后件 q 为真。则 $p\to q$ 为真，因为该命题形如 **T→T** 或 **F→T**，而它们都为真。因此，若可以证明 q 为真，则可以给出 $p\to q$ 的证明，这称为*平凡证明*。当证明定理的特殊情形（见对分情形证明的讨论）时，以及在数学归纳法（它是 3.2 节讨论的一种证明技术）中，平凡证明常常是重要的。

例 17 设 $P(n)$ 是命题“若 a 和 b 是满足 $a\geqslant b$ 的正整数，则 $a^n\geqslant b^n$”。证明命题 $P(0)$ 为真。

解 命题 $P(0)$ 是“若 $a\geqslant b$，则 $a^0\geqslant b^0$”。因为 $a^0=b^0=1$，所以 $P(0)$ 的后件为真。因此，$P(0)$ 为真。这是平凡证明法的一个例子。注意在这个证明里不需要前件，它是命题“$a\geqslant b$。” ■

假定可以找到矛盾式 q 使得 $\neg p\to q$ 为真，即 $\neg p\to$ **F** 为真。于是命题 $\neg p$ 必然为假。所以 p 必然为真。当可以找到矛盾式（比如 $r\wedge\neg r$）使得有可能证明蕴涵式 $\neg p\to(r\wedge\neg r)$ 为真时，就可以使用这种技术。这种类型的论证称为*归谬证明*。

例 18 通过给出归谬证明来证明 $\sqrt{2}$ 是无理数。

解 设 p 是命题“$\sqrt{2}$ 是无理数”。假定 $\neg p$ 为真。则 $\sqrt{2}$ 是有理数。将要证明它导致矛盾。在 $\sqrt{2}$ 是有理数的假设下，存在整数 a 和 b 满足 $\sqrt{2}=a/b$，其中 a 和 b 没有公因子（所以分数 a/b 是既约的）。因为 $\sqrt{2}=a/b$，所以当这个等式的两端都平方时，就得出 $2=a^2/b^2$。因此，$2b^2=a^2$。这意味着 a^2 是偶数，它蕴涵着 a 是偶数。另外，因为 a 是偶数，所以对某个整数 c 来说有 $a=2c$。因此，$2b^2=4c^2$，所以 $b^2=2c^2$。这意味着 b^2 是偶数。因此，b 也

必然是偶数。

已经证明了$\neg p$蕴涵着$\sqrt{2}=a/b$，其中a和b没有公因子，以及2整除a和b。这是矛盾的，因为已经证明了$\neg p$既蕴涵r又蕴涵$\neg r$，其中r是命题：a和b是没有公因子的整数。因此，$\neg p$为假，所以p："$\sqrt{2}$是无理数"为真。 ■

对一个蕴涵式的间接证明可以改写成归谬证明。在间接证明里，通过用直接证明来证明$\neg q \to \neg p$为真，来证明$p \to q$为真。即在$p \to q$的间接证明里，假定$\neg q$为真而证明$\neg p$也必然为真。为了把$p \to q$的间接证明改写成归谬证明，假定p和$\neg q$都为真。然后利用$\neg q \to \neg p$的间接证明的步骤，来证明$\neg p$也必然为真。这样就得出矛盾式$p \wedge \neg p$，由此完成归谬证明。例19说明如何把对蕴涵式的间接证明改写成归谬证明。

例19 给出定理"若$3n+2$是奇数，则n是奇数"的归谬证明。

解 假定$3n+2$是奇数而n不是奇数，所以n是偶数。按照在例15解答里的同样步骤(这个定理的间接证明)，可以证明若n是偶数则$3n+2$是偶数。这与$3n+2$是奇数的假定矛盾，证毕。 ■

为了证明形如

$$(p_1 \vee p_2 \vee \cdots \vee p_n) \to q$$

的蕴涵式，可以用重言式

$$[(p_1 \vee p_2 \vee \cdots \vee p_n) \to q] \leftrightarrow [(p_1 \to q) \wedge (p_2 \to q) \wedge \cdots \wedge (p_n \to q)]$$

来作为推理规则。这个推理规则说明，可以通过对$i=1, 2, \cdots, n$分别证明每个蕴涵式$p_i \to q$来证明由命题$p_1, p_2, \cdots, p_n$的析取式组成前件的原来的蕴涵式。这种论证称为分情形证明。有时为了证明蕴涵式$p \to q$为真，方便的做法是用析取式$p_1 \vee p_2 \vee \cdots \vee p_n$代替$p$作为蕴涵式的前件，其中$p$与$p_1 \vee p_2 \vee \cdots \vee p_n$等价。考虑下面的例子。

例20 证明蕴涵式"若n是不能被3整除的整数，则$n^2 \equiv 1 \pmod 3$。"

解 设p是命题"n不能被3整除"，设q是命题"$n^2 \equiv 1 \pmod 3$"。则p等价于$p_1 \vee p_2$，其中p_1是"$n \equiv 1 \pmod 3$"而p_2是"$n \equiv 2 \pmod 3$"。因此，为了证明$p \to q$，可以证明$p_1 \to q$和$p_2 \to q$。容易给出这两个蕴涵式的直接证明。

首先，假定p_1为真。则$n \equiv 1 \pmod 3$，所以对某个整数k来说有$n=3k+1$。因此，

$$n^2 = 9k^2 + 6k + 1 = 3(3k^2 + 2k) + 1。$$

由此得出$n^2 \equiv 1 \pmod 3$。因此，蕴涵式$p_1 \to q$为真。其次，假定p_2为真。则$n \equiv 2 \pmod 3$，所以对某个整数k来说有$n=3k+2$。因此，

$$n^2 = 9k^2 + 12k + 4 = 3(3k^2 + 4k + 1) + 1。$$

因此，$n^2 \equiv 1 \pmod 3$，所以蕴涵式$p_2 \to q$为真。

因为已经证明了$p_1 \to q$和$p_2 \to q$都为真，所以可以得出$(p_1 \vee p_2) \to q$为真。另外，因为p等价于$p_1 \vee p_2$，所以$p \to q$为真。 ■

为了证明其本身是一个等价式的定理，即形如$p \leftrightarrow q$的命题，其中p和q都是命题，可

以使用重言式

$$(p\leftrightarrow q)\leftrightarrow[(p\rightarrow q)\wedge(q\rightarrow p)]$$

即如果证明了蕴涵式“若 p 则 q” 和“若 q 则 p”，那么就可以证明命题“p 当且仅当 q”。

例 21 证明定理“整数 n 是奇数当且仅当 n^2 是奇数”。

解 这个定理是形如“p 当且仅当 q”，其中 p 是“n 是奇数”而 q 是“n^2 是奇数。”为了证明这个定理，需要证明 $p\rightarrow q$ 和 $q\rightarrow p$ 都为真。

已经证明了（在例 14 里）$p\rightarrow q$ 为真。将用间接证明来证明 $q\rightarrow p$。假定它的后件为假，即 n 是偶数。则对某个整数 k 来说有 $n=2k$。于是 $n^2=4k^2=2(2k^2)$，所以 n^2 是偶数（因为它是 2 的倍数）。这样就完成了 $q\rightarrow p$ 的间接证明。

因为已经证明了 $p\rightarrow q$ 和 $q\rightarrow p$ 都为真，所以就已经证明了这个定理为真。 ■

有时候定理说几个命题都是等价的。这样的定理说命题 p_1，p_2，p_3，…，p_n 都是等价的。这个定理可以写成

$$p_1\leftrightarrow p_2\leftrightarrow\cdots\leftrightarrow p_n$$

它说所有的 n 个命题都具有相同的真值。证明这些命题互相等价的一种方式是使用重言式

$$[p_1\leftrightarrow p_2\leftrightarrow\cdots\leftrightarrow p_n]\leftrightarrow[(p_1\rightarrow p_2)\wedge(p_2\rightarrow p_3)\wedge\cdots\wedge(p_n\rightarrow p_1)]$$

这个重言式说明，若可以证明蕴涵式 $p_1\rightarrow p_2$，$p_2\rightarrow p_3$，…，$p_n\rightarrow p_1$ 都为真，则命题 p_1，p_2，p_3，…，p_n 都是等价的。

例 22 证明当 n 是整数时，下列三个命题等价：

p_1：$n \bmod 3=1$ 或 $n \bmod 3=2$

p_2：n 不能被 3 整除

p_3：$n^2\equiv 1\pmod 3$

解 为了证明这些命题等价，证明蕴涵式 $p_1\rightarrow p_2$，$p_2\rightarrow p_3$ 和 $p_3\rightarrow p_1$ 都为真。

将用直接证明来证明 $p_1\rightarrow p_2$ 为真。假定 $n \bmod 3=1$ 或 2。根据整除算法有 $n=3q+r$，其中 $0\leqslant r<3$。根据 mod 的定义，我们有 $r\equiv n\pmod 3$。因为 n 被 3 整除当且仅当 $r=0$，所以假定 $n \bmod 3=1$ 或 2，就蕴涵着 n 不能被 3 整除。这样就完成了 $p_1\rightarrow p_2$ 的证明。

在例 20 里已经证明了 $p_2\rightarrow p_3$ 为真。

将用间接证明来证明 $p_3\rightarrow p_1$ 为真。假定这个蕴涵式的后件为假，即 $n \bmod 3$ 既不是 1 也不是 2。因为 $n \bmod 3$ 等于 0，1 或 2，所以看出 $n \bmod 3=0$。这意味着 $3\mid n$，所以对某个整数 k 来说有 $n=3k$。这蕴涵着 $n^2=9k^2=3(3k^2)$，这说明 $n^2\equiv 0\pmod 3$，所以 p_3 为假。这样就完成了 $p_3\rightarrow p_1$ 的间接证明，因而也完成了定理的证明 ■

3.1.6 定理与量词

许多定理都叙述成带量词的命题。可以使用各种方法来证明作为量词式的定理。这里将要描述一些最常见的这类方法。

许多定理都断言存在特定类型的对象。这种类型的定理是形如 $\exists xP(x)$ 的命题，其

中 P 是谓词。对形如 $\exists xP(x)$ 的命题的证明称为*存在性证明*。有多种方式来证明这种类型的定理。有时通过找出一个使得 $P(a)$ 为真的元素 a 来给出 $\exists xP(x)$ 的存在性证明。这样的存在性证明称为*构造性的*。给出非构造性的证明也是可能的；即不是找出使 $P(a)$ 为真的元素 a，而是以某种其他方式来证明 $\exists xP(x)$ 为真。给出非构造性证明的一种普通方法是使用归谬证明，证明该存在量词式的否定式蕴涵着矛盾。下列例子说明构造性的存在性证明的概念。

例 23　*构造性的存在性证明*　证明对每个正整数 n 来说都存在 n 个连续的正合数。注意这要求证明量词式：

$\forall n\ \exists x$（对 $i=1, 2, \cdots, n$ 来说有 $x+i$ 是合数）。

解　设

$$x=(n+1)!+1$$

考虑整数

$$x+1, x+2, \cdots, x+n$$

注意对 $i=1, 2, \cdots, n$ 来说，$i+1$ 整除 $x+i=(n+1)!+(i+1)$。因此，已经给出了 n 个连续的正合数。注意在这个解答里，已经产生出使得对 $i=1, 2, \cdots, n$ 来说 $x+i$ 是合数的数 x。因此，这是构造性的存在性证明的例子。■

注意　*例 23 中的证明可以在古希腊数学家欧几里德的著作中找到。下面给出非构造性证明的一个例子。*

例 24　*非构造性的存在性证明*　证明对每个正整除 n 来说都存在比 n 大的素数。这个问题要求证明一个存在量词式，即 $\exists xQ(x)$，其中 $Q(x)$ 是命题"x 是素数并且 x 比 n 大"，论域是正整数集合。

解　设 n 是正整数。为了证明存在比 n 大的素数，考虑整数 $n!+1$。因为每个整数都有素因子，所以至少存在一个整除 $n!+1$ 的素数。（一种可能性是 $n!+1$ 已经是素数了。）注意当 $n!+1$ 除以一个小于或等于 n 的整数时，余数等于 1。因此，这个整数的任何素因子都必然比 n 大。这样就证明了结果。这个论证是非构造性的存在性证明，因为没有给出比 n 大的素数。它仅仅证明了必然存在这样一个素数。■

假定形如 $\forall xP(x)$ 的一个命题为假。如何证明它？回忆一下，命题 $\neg\forall xP(x)$ 与 $\exists x\neg P(x)$ 是等价的。这就意味着，若找出一个使得 $P(a)$ 为假的元素 a，则证明了 $\exists x\neg P(x)$ 为真，而它意味着 $\forall xP(x)$ 为假。一个使得 $P(a)$ 为假的元素 a 称为*反例*。注意为了证明 $\forall xP(x)$ 为假，仅仅需要找到一个反例。

例 25　证明断言"所有素数都是奇数"为假。

解　命题"所有素数都是奇数"是全称量词式，即

$$\forall xO(x)$$

其中 $O(x)$ 是命题"x 是奇数"，论域是素数集合。注意 $x=2$ 是一个反例，因为 2 是一个偶

素数。因此，命题“所有素数都是奇数”为假。■

一种常见的错误是假定一个或多个特例就证明了命题为真。无论存在多少个使得$P(x)$为真的特例，全称量词式$\forall xP(x)$仍然可能为假。考虑下面的例子。

例 26 是否对所有非负整数n来说n^2-n+41都是素数？即命题$\forall nP(n)$是不是定理，其中$P(n)$是命题“n^2-n+41是素数”而论域是非负整数集合？

解 要确定是否对所有非负整数n来说n^2-n+41都是素数，可以首先检查是否对最小的几个非负整数来说它都是素数。检查发现对不超过 40 的所有非负整数来说n^2-n+41都是素数（读者可以验证）。不过，假如因此断定这是足够的验证，那么就得出了错误的结论。对所有非负整数来说n^2-n+41都是素数，这不是真的。当$n=41$时，它是合数（读者应当验证它）。■

例 26 有助于说明这样一个要点：即使存在许多使命题为真的特例，命题也可能不为真。

3.1.7 停机问题

现在将要描述计算机科学里最著名的定理之一的证明。将要证明：存在一个不能用任何过程来解决的问题。即像在 2.2 节指出过的那样，将要证明存在一个不可解的问题。将要研究的这个问题是停机问题。这个问题问：是否存在做下述工作的过程，这个过程的输入是一个计算机程序和这个程序的一个输入，这个过程判定当该程序用该输入运行时，程序是否最终停止。假如存在这样的过程，那么拥有它就有好处。当编写和调试程序时，假如能够测试程序是否进入死循环，这肯定是有帮助的。不过，图灵（Alan Turing）在 1936 年就证明了不存在任何这样的过程（见 10.4 节他的生平）。

在给出停机问题是不可解的证明之前，首先注意，不能通过简单地运行一个程序并观察它做什么，来确定当在给定的输入上运行时它是否停机。若这个程序停止了，则得出了答案，但是若在经过了任何固定的时间之后它仍然在运行，则不知道它究竟是永不停机，还是等待的时间还没有达到它停机的时间。无论如何，设计出一个只有在经过 10 亿年之后才停止的程序并不难。

将要描述图灵对停机问题是不可解的证明；它是归谬证明。（读者应当注意，证明不是十分严格的，因为没有明确地定义什么是过程。要弥补这个不足，就需要图灵机的概念。这个概念在 10.5 节介绍。）

证 假定停机问题有解，而且它是称为$H(P,I)$的过程。过程$H(P,I)$有两个输入，一个输入是程序P，另一个输入是I，I是程序P的输入。若H判定当给定I作为输入时P停止，则$H(P,I)$产生字符串“停机”来作为输出。否则，$H(P,I)$产生字符串“死循环”来作为输出。现在将要导出矛盾。

当把过程编码时，就把过程表示成字符串；可以把这个字符串解释成比特序列。这意味着可以用程序自身来作为数据。因此可以把一个程序当作另外一个程序的输入，甚至当作它自身的输入。因此，H可以用同一个程序P来作为它的两个输入，其中一个是程序而另外一个是程序的输入。H应当能够判定当把P自身的复制品作为P的输入时，P是否将停机。

为了证明不存在任何解决停机问题的过程H，构造一个简单的过程$K(P)$，它利用

$H(P,P)$的输出来如下的工作。若 $H(P,P)$的输出是“死循环”，这意味着当把 P 自身的复制品作为 P 的输入时 P 死循环，则$K(P)$停机。若 $H(P,P)$的输出是“停机”，这意味着当把 P 自身的复制品作为 P 的输入时 P 停机，则$K(P)$死循环。即 $K(P)$与 $H(P,P)$的输出是相反的（见图 3-1）。

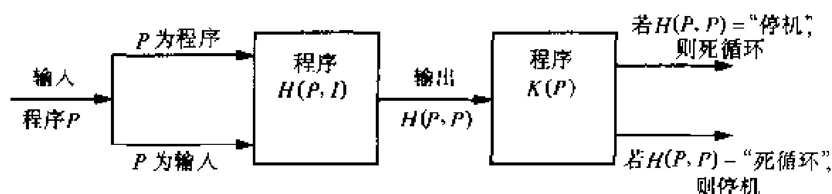


图 3-1

现在假定把 K 作为给 K 的输入。注意，若$H(K,K)$的输出是“死循环”，则根据 K 的定义，可以看出$K(K)$停机。否则，若 $H(K,K)$的输出是“停机”，则根据 K 的定义，可以看出$K(K)$死循环，这违背 H 所说出的结果。在两种情形里都有矛盾。

因此，H 不能总是给出正确的答案。所以不存在解决停机问题的过程。 □

3.1.8　关于证明的一些评注

已经描述了证明定理的各种方法。注意，在这里没有给出证明定理的算法。不存在这样的过程。

存在着许多定理，通过直接利用前提和定理里的名词的定义，就容易找出其证明。不过，通常要是不借助于灵活地利用间接证明（比如归谬证明）或一些其他的证明技术，就难以证明一个定理。构造证明是一种艺术，只有通过尝试各种使用途径才能学会它。

另外，许多似乎是定理的命题已经抗拒了数学家几百年的顽强努力。例如，像“每个大于 4 的正偶数都是两个素数之和”这个命题，就还没有被证明为真，而且也没有找出任何反例，尽管对直到 10^{14}的所有正偶数都已经验证过了。这个命题以哥德巴赫[⊖]猜想而闻名，它是在数学里许多真值未知的断言之一。

练习

1. 在下列每个论证里使用了什么推理规则？
   a）爱丽丝主修数学。因此，爱丽丝主修数学或计算机科学。
   b）杰瑞主修数学和计算机科学。因此，杰瑞主修数学。
   c）若天气下雨，则游泳池将关闭。天气下雨。因此，游泳池关闭。
   d）若今天下雪，则大学将关闭。今天大学没有关闭。因此，今天没有下雪。


⊖ 哥德巴赫（Christian Goldbach，1690—1764）　哥德巴赫诞生在普鲁士的哥尼斯堡，这个城市以著名的七桥问题而闻名于世（在 7.5 节将研究这个问题）。他在 1725 年成为在圣彼得堡的科学院的数学教授。哥德巴赫在 1728 年来到莫斯科当沙皇儿子的家庭教师。当他在 1741 年成为俄国外交部的职员时，他进入了政界。哥德巴赫最出名的事情是他与包括欧拉和伯努利在内的杰出数学家们的通信，他在数论里的著名猜想，以及对数学分析的许多贡献。

e) 若我去游泳，则我将在太阳下停留过久。若我在太阳下停留过久，则我将有晒斑。因此，若我去游泳，则我将有晒斑。

2. 在下列每个论证里使用了什么推理规则？
   a) 袋鼠生活在澳大利亚并且是有袋动物。因此，袋鼠是有袋动物。
   b) 今天气温高于 100 度或污染是有害的。今天户外气温低于 100 度。因此，污染是有害的。
   c) 琳达是优秀的游泳者。若琳达是优秀的游泳者，则她可以当救生员。因此，琳达可以当救生员。
   d) 今年夏天史蒂夫将在计算机公司工作。因此，今年夏天史蒂夫将在计算机公司工作或者在海滩闲逛。
   e) 若我整夜地做这个家庭作业，则我可以解答所有的习题。若我解答所有的习题，则我将理解这些材料。因此，若我整夜地做这个家庭作业，则我将理解这些材料。
3. 使用推理规则构造一个论证来证明前提"兰迪努力地工作"，"若兰迪努力地工作，则他是一个笨孩子"，和"若兰迪是一个笨孩子，则他将得不到这个工作"蕴涵着结论"兰迪将得不到这个工作"。
4. 使用推理规则构造一个论证来证明前提"若天气不下雨或天气不起雾，则航行比赛将举行而且救生表演将进行"，"若航行比赛举行，则将颁发奖品"，而"没有颁发奖品"蕴涵着结论"天气下雨"。
5. 在下面的著名论证里使用了什么推理规则？"所有的人都是要死的。苏格拉底是人。因此，苏格拉底是要死的"。
6. 在下面的论证里使用了什么推理规则？"没有人是岛屿。曼哈顿是岛屿。因此，曼哈顿不是人。"
7. 对下列的每组前提，可以得出什么样的相关结论或一组结论？解释从前提获得每个结论所使用的推理规则。
   a) "若我在某一天休息，则那天下雨或下雪"。"我在周二休息或在周四休息"。"周二出太阳"。"周四未下雪。"
   b) "若我吃了辣的食物，则我做奇怪的梦"。"若当我睡觉时有雷声，则我做奇怪的梦"。"我没有做奇怪的梦。"
   c) "我聪明或幸运"。"我不幸运"。"若我幸运，则我将赢得抽奖。"
   d) "每个主修计算机科学的人都有个人电脑"。"拉尔夫没有个人电脑"。"安妮有个人电脑。"
   e) "对公司有利的就对美国有利"。"对美国有利的就对你有利"。"对公司有利的就是你购买许多东西。"
   f) "所有的啮齿类动物都啃它们的食物。""老鼠是啮齿类动物"。"野兔不啃它们的食物"。"蝙蝠不是啮齿类动物"。
8. 对下列的每组前提，可以得出什么样的相关结论或一组结论？解释从前提获得每个结论所使用的推理规则。
   a) "若我打曲棍球，则我第二天感到酸痛"。"若我感到酸痛，则我用水疗"。"我没有用水疗"。

b) “若我工作，则天气晴或半晴”。“我上周一工作或上周五工作”。“周二天气不晴”。“周五天气不是半晴。”

c) “所有的昆虫都有六条腿”。“蜻蜓是昆虫”。“蜘蛛不是六条腿”。“蜘蛛吃蜻蜓”。

d) “每个学生都有因特网账号”。“荷马没有因特网账号”。“马奇有因特网账号”。

e) “所有对健康有益的食物都不好吃”。“豆腐对健康有益”。“你只吃好吃的东西”。“你不吃豆腐”。“汉堡包对健康无益”。

f) “我在做梦或在幻觉中”。“我不在做梦”。“若我在幻觉中，则我看见大象在路上跑”。

9. 对下列每个论证，解释对每个步骤使用了哪条推理规则。

a) “本班学生道格知道如何用 JAVA 写程序。知道如何用 JAVA 写程序的每个人都可以得到高薪的工作。因此，本班的某个人可以得到高薪的工作。”

b) “本班的某个人喜欢观察鲸鱼。每个喜欢观察鲸鱼的人都关心海洋污染。因此，本班里有某个人关心海洋污染。”

c) “本班的 93 个学生每人拥有一台个人电脑。拥有个人电脑的每个人都能使用字处理软件。因此，本班学生泽克能使用字处理软件。”

d) “新泽西的每个人都生活在距离海洋 50 英里之内。新泽西的某个人从来没有见过海洋。因此，生活在距离海洋 50 英里之内的某个人从来没有见过海洋。”

10. 对下列每个论证，解释对每个步骤使用了哪条推理规则。

a) “本班学生琳达拥有红色折篷汽车。拥有红色摺篷汽车的每个人都至少领到一张超速罚单。因此，本班的某个人领到一张超速罚单。”

b) “五位室友中的每一位——梅丽莎、阿茸、拉尔夫、维妮莎、和基绍恩——都学习了离散数学课程。每个学习了离散数学课程的学生都可以学习算法课程。因此，所有的五位室友明年都可以学习算法课程。”

c) “塞雷斯 (John sayles) 制作的所有电影都好看。塞雷斯制作过关于煤矿工人的电影。因此，存在好看的关于煤矿工人的电影。”

d) “本班有人到过法国。到过法国的每个人都访问过卢浮宫。因此，本班有人访问过卢浮宫。”

11. 判定下列每个论证是否有效。若论证是正确的，则使用了什么推理规则？若它不正确，则发生了什么谬误？

a) 若 n 是实数使得 $n>1$，则 $n^2>1$。假定 $n^2>1$。于是 $n>1$。

b) 若数 $\log_2 3$ 不是两个整数的商，则它是无理数。因此，因为 $\log_2 3$ 不能写成 a/b 的形式，其中 a 和 b 是整数，所以它是无理数。

c) 若 n 是使 $n>3$ 的实数，则 $n^2>9$。假定 $n^2\leqslant 9$。于是 $n\leqslant 3$。

d) 一个正整数是完全平方数或者它有偶数个正整数因子。假定 n 是有奇数个正整数因子的正整数。于是 n 是完全平方数。

e) 若 n 是使 $n>2$ 的实数，则 $n^2>4$。假定 $n\leqslant 2$。于是 $n^2\leqslant 4$。

12. 下列论证是对定理“若 n^2 不能被 3 整除，则 n 不能被 3 整除”的不正确的证明。它不正确的原因是使用了循环论证。在推理中何处犯了错误？

若 n^2 不能被 3 整除，则对某个整数 k 来说 n^2 不等于 $3k$。所以，对某个整数 l 来说 n 不等于 $3l$。因此，n 不能被 3 整除。

13. 证明命题$P(0)$，其中$P(n)$是命题“若n是个大于1的正整数，则$n^2>n$”。你使用什么类型的证明？
14. 证明命题$P(1)$，其中$P(n)$是命题“若n是个正整数，则$n^2>n$”。你使用什么类型的证明？
15. 设$P(n)$是命题“若a和b是正实数，则$(a+b)^n \geq a^n+b^n$”。证明$P(1)$为真。你使用什么类型的证明？
16. 证明：偶数的平方是偶数，使用
    a) 直接证明。
    b) 间接证明。
    c) 归谬证明。
17. 证明：若n是整数而且n^3+5是奇数，则n是偶数，使用
    a) 间接证明。
    b) 归谬证明。
18. 证明：若n是整数而且$3n+2$是偶数，则n是偶数，使用
    a) 间接证明。
    b) 归谬证明。
19. 证明：两个奇数之和是偶数。
20. 证明：两个有理数之和是有理数。
21. 证明：一个无理数与一个有理数之和是无理数，使用归谬证明。
22. 证明：两个有理数之积是有理数。
23. 证明或反驳：两个无理数之积是无理数。
24. 证明或反驳：一个非零有理数与一个无理数之积是无理数。

*25. 证明或反驳：每当n是正整数$n^2-79n+1601$就是素数。

26. 证明或反驳：对所有非负整数n来说2^n+1都是素数。
27. 证明：$\sqrt[3]{3}$是无理数。

*28. 证明：若n是正整数但不是完全平方数，则$\sqrt{n}$是无理数。

29. 证明：若x和y都是实数，则$\max(x,y)+\min(x,y)=x+y$。[提示：使用分情形证明，两种情形分别对应于$x \geq y$和$x<y$。]
30. 证明：当一个不能被5整除的整数的平方除以5时，余数是1或4。[提示：使用分情形证明，各种情形对应于整数除以5的可能的余数。]
31. 证明：若x和y都是实数，则$|x|+|y| \geq |x+y|$（其中$|x|$表示x的绝对值，若$x \geq 0$则它等于x，若$x \leq 0$则它等于$-x$）。
32. 使用分情形证明来证明：对所有整数n来说都有$\lfloor n/2 \rfloor \lceil n/2 \rceil = \lfloor n^2/4 \rfloor$。
33. 使用分情形证明来证明：每当a，b和c都是实数时就有$\min(a,\min(b,c))=\min(\min(a,b),c)$。
34. 证明：若n是正整数，则n是偶数当且仅当$7n+4$是偶数。
35. 证明：若n是正整数，则n是奇数当且仅当$5n+6$是奇数。
36. 证明：$m^2=n^2$当且仅当$m=n$或$m=-n$。

*37. 设p是素数。证明：$a^2 \equiv b^2 \pmod p$当且仅当$a \equiv b \pmod p$或$a \equiv -b \pmod p$。

38. 证明或反驳：每当 n 是大于 1 的正整数就有 n^2-1 是合数。
39. 证明或反驳：若 m 和 n 是整数使得 $mn=1$，则 $m=1$ 并且 $n=1$，或者 $m=-1$ 并且 $n=-1$。
40. 证明或反驳：每当 m 是正整数时就有 $a \bmod m + b \bmod m = (a+b) \bmod m$。
41. 证明或反驳：每个正整数都可以写成两个整数的平方之和。
42. 证明：若 n 是正整数使得它的因子之和是 $n+1$，则 n 是素数。你使用什么类型的证明？
43. 证明或反驳下面每个关于下取整和上取整函数的命题。
    a) 对所有实数 x 来说 $\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$。
    b) 每当 x 是实数时就有 $\lfloor 2x \rfloor = 2\lfloor x \rfloor$。
    c) 每当 x 和 y 是实数时就有 $\lceil x \rceil + \lceil y \rceil - \lceil x+y \rceil = 0$ 或 1。
    d) 对所有实数 x 和 y 来说都有 $\lceil xy \rceil = \lceil x \rceil \lceil y \rceil$。
    e) 对所有实数 x 来说都有 $\left\lfloor \frac{x}{2} \right\rfloor = \left\lceil \frac{x+1}{2} \right\rceil$。
44. 证明或反驳下面每个关于下取整和上取整函数的命题。
    a) 对所有实数 x 来说 $\lfloor \lceil x \rceil \rfloor = \lceil x \rceil$。
    b) 对所有实数 x 和 y 来说 $\lfloor x+y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$。
    c) 对所有实数 x 来说 $\lceil \lceil x/2 \rceil /2 \rceil = \lceil x/4 \rceil$。
    d) 对所有实数 x 来说 $\lfloor \sqrt{\lceil x \rceil} \rfloor = \lfloor \sqrt{x} \rfloor$。
    e) 对所有实数 x 和 y 来说 $\lfloor x \rfloor + \lfloor y \rfloor + \lfloor x+y \rfloor \leqslant \lfloor 2x \rfloor + \lfloor 2y \rfloor$。
45. 证明：若 x 是正实数，则
    a) $\lfloor \sqrt{\lfloor x \rfloor} \rfloor = \lfloor \sqrt{x} \rfloor$
    b) $\lceil \sqrt{\lceil x \rceil} \rceil = \lceil \sqrt{x} \rceil$
46. 证明：若 m 和 n 都是正整数而 x 是正实数，则 $\left\lfloor \frac{\lfloor x \rfloor + n}{m} \right\rfloor = \left\lfloor \frac{x+n}{m} \right\rfloor$。

*47. 证明：若 m 是正整数而 x 是正实数，则

$$\lfloor mx \rfloor = \lfloor x \rfloor + \left\lfloor x + \frac{1}{m} \right\rfloor + \left\lfloor x + \frac{2}{m} \right\rfloor + \cdots + \left\lfloor x + \frac{m-1}{m} \right\rfloor$$

**48. 证明：若 a 和 b 是正无理数使得 $1/a + 1/b = 1$，则每个正整数都可以惟一地表示成 $\lfloor ka \rfloor$ 或 $\lfloor kb \rfloor$，其中 k 是某个正整数。

49. 证明：在实数 $a_1, a_2, \cdots, a_n$ 中至少有一个数大于或等于这些数的平均值。你使用什么类型的证明？

*50. 使用练习 49 来证明：若把前 10 个正整数以任意顺序放在一个圆周上，则圆周上存在位置相邻的 3 个整数，它们之和大于或等于 17。

51. 证明：若 n 是整数，则下面 4 个命题等价：(i) n 是偶数；(ii) $n+1$ 是奇数；(iii) $3n+1$ 是奇数；(iv) $3n$ 是偶数。
52. 证明：若 n 是整数，则下面 3 个命题等价：(i) 5 整除 n；(ii) 5 整除 n^2；(iii) $n^2 \not\equiv \pm 1 \pmod 5$。
53. 证明或反驳：存在 3 个连续的正奇数都是素数，即形如 p，$p+2$ 和 $p+4$ 的奇素数。

54. 证明或反驳：给定正整数 n，存在 n 个连续的正奇数都是素数。
55. 哪些推理规则被用来证明在1.3节练习20里所描述的卡洛尔（Lewis Carroll）的论证的结论？
56. 哪些推理规则被用来证明在1.3节练习21里所描述的卡洛尔（Lewis Carroll）的论证的结论？
57. 给出命题“对每个正整数 n 来说都存在可被 n 个以上素数整除的整数”的构造性证明。
58. 找出命题“对每个素数 n 来说都有 $n+2$ 是素数”的反例。

*59. 证明：存在无穷多个素数与3模4同余。你的证明是构造性的还是非构造性的？[提示：一种方法是假定仅存在有穷多个这样的素数 $p_1, p_2, \cdots, p_n$。设 $q=4p_1p_2\cdots p_n+3$。证明 q 必有与3模4同余的素因子，它不在 n 个素数 $p_1, p_2, \cdots, p_n$ 之中。]

60. 证明或反驳：若 p_1，p_2，…，p_n 是 n 个最小的素数，则 $p_1p_2\cdots p_n+1$ 是素数。
61. 证明：可以通过证明蕴涵式 $p_1 \to p_4$，$p_3 \to p_1$，$p_4 \to p_2$，$p_2 \to p_5$ 和 $p_5 \to p_1$ 来证明命题 p_1，p_2，p_3，p_4 和 p_5 等价。
62. 证明或反驳：若 a 和 b 都是有理数，则 a^b 也是有理数。
63. 证明：存在无理数 a 和 b 使得 a^b 是有理数。你的证明是构造性的还是非构造性的？[提示：设 $a=\sqrt{2}$ 和 $b=\sqrt{5}$。证明：a^b 或（a^b）b 是有理数。]
64. 证明：用多米诺骨牌（1×2大小）可以完全覆盖8×8的棋盘。

*65. 证明：不可能用多米诺骨牌完全覆盖切掉两个对角格子的8×8的棋盘。

*66. 逻辑问题，来自正(如 WFF'N PROOF)中的逻辑游戏，有下面两个假设：
   1）“逻辑是困难的或没有许多学生喜欢逻辑”。
   2）“若数学是容易的，则逻辑不是困难的”。

   把这些假设翻译成包括命题变元和逻辑联结词的命题，判定下面每个命题是不是这些前提的有效结论：

   a）若有许多学生喜欢逻辑，则数学是容易的。
   b）若数学不是容易的，则没有许多学生喜欢逻辑。
   c）数学不是容易的或逻辑是困难的。
   d）逻辑不是困难的或数学不是容易的。
   e）若没有许多学生喜欢逻辑，则数学不是容易的或逻辑不是困难的。

*67. 判定下列论证（来自Back house[Ba86]）是否有效：

> 假如超人能够并愿意防止邪恶，则他将这样做。假如超人不能够防止邪恶，则他将是无能的；假如超人不愿意防止邪恶，则他将是恶意的。超人没有防止邪恶。若超人存在，则他是无能的或恶意的。因此，超人不存在。

*消解法*是一种在人工智能和机器证明程序里广泛地使用的证明方法。消解法的基础是从重言式 $((p \vee q) \wedge (\neg p \vee r)) \to (q \vee r)$ 导出的推理规则。

68. 使用消解法推理规则来证明命题“你将赢得抽奖或你将被提升”，给定的前提是“你将辞职或你将赢得抽奖”，“你将不辞职或你将找个更好的工作”，和“你将不找个更好的工作或你将被提升”。

3.2 数学归纳法

3.2.1 引言



前 n 个正奇数之和的公式是什么？对 $n=1$，2，3，4，5 来说前 n 个正奇数之和是

$1=1$，　　$1+3=4$，　　$1+3+5=9$，

$1+3+5+7=16$，　　$1+3+5+7+9=25$

根据这些值，有理由猜测前 n 个正奇数之和是 n^2。假如事实上这个猜测是正确的，我们就需要一种方法来证明这个猜测是正确的。

数学归纳法是证明这种类型的断言的极为重要的证明技术。在本节和下面几章里，将要看到数学归纳法被广泛地用来证明关于各种各样离散对象的结果。例如用来证明关于下列内容的结果：算法的复杂性，某些类型计算机程序的正确性，关于图与树的定理，以及各种等式和不等式等。

在本节里将要描述如何使用数学归纳法和为什么它是有效的证明技术。极端重要的是注意：数学归纳法只能证明通过其他方式所获得的结果。它不是发现公式或定理的工具。

3.2.2 良序性

数学归纳法的有效性来源于下面这个关于整数集的基本公理。

良序性　每个非空的非负整数集都有最小元。

良序性经常可以直接地用在证明里。

例1　用良序性证明整除算法。回忆一下，整除算法说：若 a 是整数而且 d 是正整数，则存在唯一的整数 q 和 r 满足 $0\leqslant r<d$ 和 $a=dq+r$。

解　设 S 是形如 $a-dq$ 的非负整数的集合，其中 q 是整数。这个集合非空，因为 $-dq$ 可以任意大（取 q 是绝对值很大的负整数）。根据良序性，S 有最小元 $r=a-dq_0$。

整数 r 非负而且 $r<d$。若不是这样，则 S 里存在更小的非负整数，即 $a-d(q_0+1)$。为了看出这一点，假设 $r\geqslant d$。因为 $a=dq_0+r$，所以 $a-d(q_0+1)=a-dq_0-d=r-d\geqslant 0$。因此，存在满足 $0\leqslant r<d$ 的整数 r 和 q。证明 q 和 r 都是唯一的，留给读者作为习题。 ■

3.2.3 数学归纳法

许多定理说：对所有正整数 n 来说 $P(n)$ 为真，其中 $P(n)$ 是命题函数，比方说是命题 $1+2+\cdots+n=n(n+1)/2$ 或命题 $n\leqslant 2^n$。数学归纳法是证明这种类型的定理的技术。换句话说，数学归纳法用来证明形如 $\forall nP(n)$ 的命题，其中论域是正整数集。

通过数学归纳法证明对每个正整数 n 来说 $P(n)$ 为真，一共包含两个步骤：

1. *基础步骤*。证明命题 $P(1)$ 为真。

2. *归纳步骤*。证明对每个正整数 n 来说蕴涵式 $P(n)\rightarrow P(n+1)$ 为真。

这里对固定的正整数 n 来说命题 $P(n)$ 称为归纳假设。当完成了数学归纳法证明的两个步骤时，就证明了对所有正整数 n 来说 $P(n)$ 为真；即证明了 $\forall nP(n)$ 为真。

把这种证明技术表示成推理规则就是

$$[P(1)\land\forall n(P(n)\to P(n+1))]\to\forall nP(n)$$

由于数学归纳法是如此重要的证明技术，所以值得详细地解释一下使用这个技术的证明步骤。为了证明对所有正整数 n 来说 $P(n)$ 为真,首先证明 $P(1)$ 为真。这等于证明当在 $P(n)$ 里用 1 替换 n 时所得到的特殊命题为真。然后必须证明对每个正整数 n 来说都有 $P(n)\to P(n+1)$ 为真。为了证明对每个正整数 n 来说这个蕴涵式为真，需要证明当 $P(n)$ 为真时 $P(n+1)$ 不能为假。可以通过假设 $P(n)$ 为真,而且证明在此假设下 $P(n+1)$ 也必然为真，来完成这个证明。

注意 *在数学归纳法证明里并不假定对所有正整数来说 $P(n)$ 为真！只是证明了：若假定 $P(n)$ 为真,则 $P(n+1)$ 也为真。因此，数学归纳法证明不属于回避问题或循环论证的情形。*

当用数学归纳法来证明定理时，首先证明 $P(1)$ 为真。然后知道 $P(2)$ 为真,因为 $P(1)$ 蕴涵 $P(2)$。另外，还知道 $P(3)$ 为真,因为 $P(2)$ 蕴涵 $P(3)$。以这样的方式继续下去，就可以看出对任意正整数 k 来说 $P(k)$ 为真。

存在多种有用的对数学归纳法的解释，它们可以帮助读者记忆这个原理是如何工作的。其中一种解释包含一队人员，人员一、人员二等等。人员一被告知一个秘密，每一个听到这个秘密的人都把它告诉队伍里的下一个人。于是 $P(1)$ 为真,因为人员一被告知了这个秘密;$P(2)$ 为真,因为人员一把这个秘密告诉了人员二;$P(3)$ 为真，因为人员二把这个秘密告诉了人员三；等等。根据数学归纳法的原理，队伍中的每个人都知道这个秘密。在图 3-2 里对此进行解释。（当然，假定每个人都原封不动地把秘密传达给下一个人，这个假设通常是不符合实际生活的。）

另一种解释数学归纳法的方式是考虑无穷长的一行多米诺骨牌，标记成 1，2，3，…，n，其中每张多米诺骨牌都直立着。设 $P(n)$ 是命题：多米诺骨牌 n 被撞倒。如果第一张多米诺骨牌被撞倒，即 $P(1)$ 为真，并且若每当第 n 张多米诺骨牌被撞倒时，它也撞倒第 $n+1$ 张多米诺骨牌，即若 $P(n)\to P(n+1)$ 为真，那么所有的多米诺骨牌都被撞倒。在图 3-3 里对此进行解释。

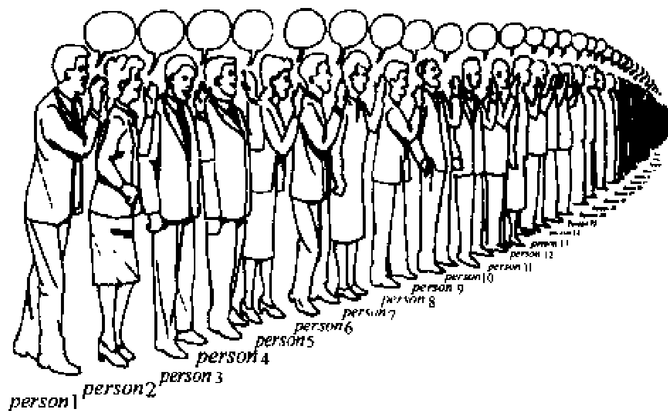


图 3-2 传说秘密的人们

为什么数学归纳法是有效的 为什么数学归纳法是一种有效的证明技术？原因来源于良序性。假定知道 $P(1)$ 为真，而且对所有正整数 n 来说命题 $P(n)\to P(n+1)$ 为真。为了证明对所有正整数来说 $P(n)$ 都为真，假定至少存在一个使 $P(n)$ 为假的正整数。那么使 $P(n)$ 为假的正整数集合 S 非空。因此，根据良序性，S 有一个最小元，把它表示成 k。可以知道 k 不是 1，因为 $P(1)$ 为真。因为 k 是正的而且大于 1，所以 $k-1$ 是一个正整数。另外，因为

$k-1$ 小于 k，它不属于 S，所以 $P(k-1)$ 必然为真。因为蕴涵式 $P(k-1)\rightarrow P(k)$ 也为真，所以实际情形必然是 $P(k)$ 为真。这与对 k 的选择相矛盾。因此，对每个正整数 n 来说 $P(n)$ 必然为真[○]。

3.2.4 数学归纳法证明的例子

将要用各种例子来说明如何用数学归纳法证明定理。首先证明前 n 个正奇数之和的公式。(在本节里用数学归纳法证明的许多定理都可以用不同的方法来证明。不过，用多种方式来证明一个定理是值得尝试的，因为一种证明方法可能成功而另外一种方法或许不成功。)

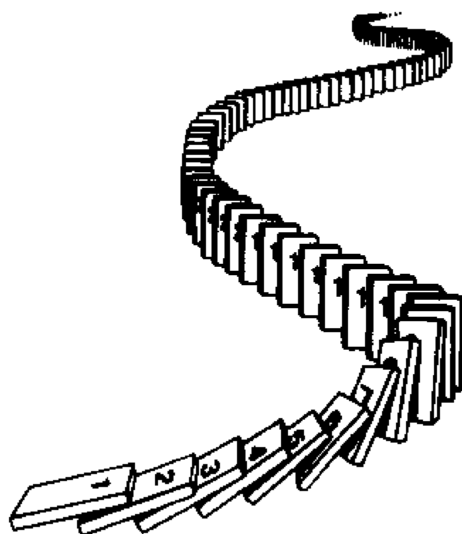


图 3-3 用多米诺骨牌解释数学归纳法如何工作

例 2 用数学归纳法证明：前 n 个正奇数之和是 n^2。

解 设 $P(n)$ 表示命题：前 n 个正奇数之和是 n^2。必须首先完成基础步骤；即必须证明 $P(1)$ 为真。然后必须完成归纳步骤；即必须证明当假定 $P(n)$ 为真时 $P(n+1)$ 为真。

基础步骤：$P(1)$ 说：前 1 个正奇数之和是 1^2。这是真的，因为第 1 个正奇数是 1。

归纳步骤：为了完成归纳步骤，必须证明对每个正整数 n 来说命题 $P(n)\rightarrow P(n+1)$ 为真。为了做到这一点，假定对正整数 n 来说 $P(n)$ 为真；即

$$1+3+5+\cdots+(2n-1)=n^2。$$

[注意第 n 个正奇数是 $(2n-1)$，因为这个整数是通过向 1 加 $n-1$ 次 2 所获得的。] 必须证明假定 $P(n)$ 为真，则 $P(n+1)$ 为真。注意 $P(n+1)$ 是命题

$$1+3+5+\cdots+(2n-1)+(2n+1)=(n+1)^2。$$

所以，假定 $P(n)$ 为真，得出

$$\begin{aligned}1+3+5+\cdots+(2n-1)+(2n+1)&=[1+3+5+\cdots+(2n-1)]+(2n+1)\\&=n^2+(2n+1)\\&=n^2+2n+1\\&=(n+1)^2\end{aligned}$$

○ 历史注记：已知的最早的对数学归纳法的使用，是在 16 世纪数学家毛洛利可（Francesco maurolico，1494—1575）的著作里。毛洛利可写过大量的关于古典数学的著作，并且对几何学和光学做出过许多贡献。在他的著作《算书二》（*Arithmeticorum Libri Duo*）里，毛洛利可给出了整数的各种性质和对这些性质的证明。为了证明其中的某些性质，他设计出数学归纳法这个方法。在这本书里他对数学归纳法的第一次使用是为了证明前 n 个正奇数之和等于 n^2。

这样证明了从$P(n)$得出$P(n+1)$。注意在第二个等式里使用了归纳假设$P(n)$，以便用n^2来代替前n个正奇数之和。

因为$P(1)$为真，而且对所有正整数n来说蕴涵式$P(n)\to P(n+1)$为真，所以数学归纳法原理就证明了对所有正整数n来说$P(n)$为真。■

下一个例子用数学归纳法来证明一个不等式。

例 3 用数学归纳法证明：对所有正整数n来说有不等式

$$n<2^n$$

解 设$P(n)$是命题“$n<2^n$”。

基础步骤：$P(1)$为真，因为$1<2^1=2$。

归纳步骤：假定对正整数n来说$P(n)$为真；即假定$n<2^n$。需要证明$P(n+1)$为真。即需要证明$n+1<2^{n+1}$。对$n<2^n$的两边加1，然后注意$1\leqslant 2^n$，这样就给出

$$n+1<2^n+1\leqslant 2^n+2^n=2^{n+1}$$

在$P(n)$为真的假定的基础上，已经证明了$P(n+1)$为真，即$n+1<2^{n+1}$。归纳步骤完毕。

因此，根据数学归纳法原理，已经证明对所有正整数n来说$n<2^n$为真。■

例 4 用数学归纳法证明：每当n是正整数时n^3-n就被3整除。

解 为了构造这个证明，设$P(n)$是命题“n^3-n被3整除”。

基础步骤：$P(1)$为真，因为$1^3-1=0$被3整除。

归纳步骤：假定$P(n)$为真；即n^3-n被3整除。必须证明$P(n+1)$为真。即必须证明$(n+1)^3-(n+1)$被3整除。注意

$$\begin{aligned}(n+1)^3-(n+1)&=(n^3+3n^2+3n+1)-(n+1)\\&=(n^3-n)+3(n^2+n)\end{aligned}$$

因为在这个和里的两项都被3整除（第一项是根据归纳步骤的假设，第二项是因为它是一个整数的3倍），由此得出$(n+1)^3-(n+1)$也被3整除。这样就完成了归纳步骤。因此，根据数学归纳法原理，每当n是正整数时n^3-n就被3整除。■

有时需要证明对$n=k$，$k+1$，$k+2$，…来说$P(n)$为真，其中k是不等于1的整数。只要改变基础步骤，就可以用数学归纳法来完成这个证明。例如，考虑例5，它证明对所有非负整数来说一个求和公式是有效的，所以需要证明对$n=0$，1，2，…来说$P(n)$为真。

例 5 用数学归纳法证明：对所有非负整数来说

$$1+2+2^2+\cdots+2^n=2^{n+1}-1$$

解 设$P(n)$是命题：对整数n来说这个公式正确。

基础步骤：$P(0)$为真，因为$2^0=1=2^1-1$。

归纳步骤：假定$P(n)$为真。为了利用这个假定完成归纳步骤，必须证明$P(n+1)$为真，即

$$1+2+2^2+\cdots+2^n+2^{n+1}=2^{(n+1)+1}-1=2^{n+2}-1$$

使用归纳假设$P(n)$，得出

$$\begin{aligned}1+2+2^2+\cdots+2^n+2^{n+1}&=(1+2+2^2+\cdots+2^n)+2^{n+1}\\&=(2^{n+1}-1)+2^{n+1}\\&=2\cdot2^{n+1}-1\\&=2^{n+2}-1\end{aligned}$$

这样就完成了归纳步骤，证毕。■

例5说明，为了用数学归纳法证明对$n=k$，$k+1$，$k+2$，…来说$P(n)$为真，其中k是不等于1的整数，就证明$P(k)$为真（基础步骤），然后证明对$n=k$，$k+1$，$k+2$，…来说蕴涵式$P(n)\to P(n+1)$为真（归纳步骤）。注意k可以是负的、零或正的。根据前面用过的多米诺骨牌的比喻，想象首先撞倒第k张多米诺骨牌（基础步骤），当每张多米诺骨牌倒下时，它就撞倒下一张多米诺骨牌（归纳步骤）。留给读者去证明这种形式的归纳是有效的（见练习68）。

例5里给出的公式是几何级数各项之和的一般结果的特殊情形，几何级数是形如a，ar，ar^2，…，ar^n，…的序列，其中a和r是实数。例如，例5里的序列是$a=1$和$r=2$的几何级数。同理，序列3，15，75，…，$3\cdot5^n$，…是$a=3$和$r=5$的几何级数。下一个例子给出这样的序列的前$n+1$项之和的公式。对这个一般公式的证明将使用数学归纳法。

例6 *几何级数求和* 用数学归纳法证明几何级数有穷多项之和的下列公式：

$$\sum_{j=0}^{n}ar^j=a+ar+ar^2+\cdots+ar^n=\frac{ar^{n+1}-a}{r-1}，\text{当 } r\neq1 \text{ 时}$$

解 为了用数学归纳法来证明这个公式，设$P(n)$是命题：这个公式里的几何级数前$n+1$项之和是正确的。

基础步骤：$P(0)$为真，因为$a=\dfrac{ar-a}{r-1}$。

归纳步骤：假定$P(n)$为真。即假定

$$a+ar+ar^2+\cdots+ar^n=\frac{ar^{n+1}-a}{r-1}$$

为了证明它蕴涵$P(n+1)$为真，对这个等式的两边加上ar^{n+1}得到

$$a+ar+ar^2+\cdots+ar^n+ar^{n+1}=\frac{ar^{n+1}-a}{r-1}+ar^{n+1}$$

改写这个等式的右边就说明

$$\frac{ar^{n+1}-a}{r-1}+ar^{n+1}=\frac{ar^{n+1}-a}{r-1}+\frac{ar^{n+2}-ar^{n+1}}{r-1}=\frac{ar^{n+2}-a}{r-1}$$

把这些等式组合起来就给出

$$a+ar+ar^2+\cdots+ar^n+ar^{n+1}=\frac{ar^{n+2}-a}{r-1}$$

这样就证明了若$P(n)$为真则$P(n+1)$必然也为真。由此完成了归纳论证并且证明了几何级数各项之和的公式是正确的。 ■

前面指出过，例5里的公式是例6里的公式当$a=1$和$r=2$时的情形。读者应当验证在一般公式里设a和r取这些值就给出与在例5里同样的公式。

在下一个例子里将证明一组正整数的倒数之和的一个重要不等式。

例7 调和数不等式 对$k=1,2,3,\cdots$来说，把调和数H_k定义成

$$H_k=1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{k}$$

例如，

$$H_4=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}=\frac{25}{12}$$

用数学归纳法证明：每当n是非负整数时就有

$$H_{2^n}\geqslant 1+\frac{n}{2}$$

解 为了完成这个证明，设$P(n)$是命题：$H_{2^n}\geqslant 1+n/2$。

基础步骤：$P(0)$为真，因为$H_{2^0}=H_1=1\geqslant 1+0/2$。

归纳步骤：假定$P(n)$为真，所以$H_{2^n}\geqslant 1+n/2$。必须证明在这个假定之下$P(n+1)$也必然为真，$P(n+1)$说$H_{2^{n+1}}\geqslant 1+(n+1)/2$。这是可以证明的，因为

$$\begin{aligned}
H_{2^{n+1}}&\geqslant 1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{2^n}+\frac{1}{2^n+1}+\cdots+\frac{1}{2^{n+1}}\\
&=H_{2^n}+\frac{1}{2^n+1}+\cdots+\frac{1}{2^{n+1}}\\
&\geqslant\left(1+\frac{n}{2}\right)+\frac{1}{2^n+1}+\cdots+\frac{1}{2^{n+1}}\quad\text{（根据归纳假设）}\\
&\geqslant\left(1+\frac{n}{2}\right)+2^n\cdot\frac{1}{2^{n+1}}\quad\text{（因为有 }2^n\text{ 项都不小于 }1/2^{n+1}\text{）}\\
&\geqslant\left(1+\frac{n}{2}\right)+\frac{1}{2}\\
&=1+\frac{n+1}{2}
\end{aligned}$$

这样就完成了证明的归纳步骤。因此，对所有非负整数n来说调和数不等式都是有效的。 ■

注意 可以用这里证明的不等式去证明调和级数

$$1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}+\cdots$$

是发散的无穷级数。在无穷级数的研究里这是一个重要的例子。

下一个例子说明如何用数学归纳法去验证有穷集合的子集个数的公式。

例8 有穷集合的子集个数 用数学归纳法证明：若S是有n个元素的有穷集合，则S

有 2^n 个子集。(在第 4 章里我们将以多种方式直接地证明这个结果。)

解 设 $P(n)$ 是命题：有 n 个元素的集合有 2^n 个子集。

基础步骤：$P(0)$ 为真，因为有 0 个元素的集合，即空集，恰有 $2^0=1$ 个子集，因为它有一个子集，即它自身。

归纳步骤：假定 $P(n)$ 为真，即每个有 n 个元素的集合都有 2^n 个子集。必须证明在这个假定下 $P(n+1)$ 也必然为真，$P(n+1)$ 是命题：每个有 $n+1$ 个元素的集合都有 2^{n+1} 个子集。为了证明它，设 T 是有 $n+1$ 个元素的集合。于是可以写出 $T=S\cup\{a\}$，其中 a 是 T 中的一个元素而 $S=T-\{a\}$。可以用下面的方式来获得 T 的子集。对 S 的每个子集 X 来说，恰存在两个 T 的子集，即 X 和 $X\cup\{a\}$。(在图 3-4 里对此进行解释。)这些集合构成了 T 的所有子集并且都互不相等。因为 S 有 2^n 个子集，所以 T 有 $2\cdot 2^n=2^{n+1}$ 个子集。这样就完成了归纳论证。 ■

例 9 证明：若 n 是正整数，则 $1+2+\cdots+n=n(n+1)/2$。

解 设 $P(n)$ 是命题：前 n 个正整数之和是 $n(n+1)/2$。必须做两件事情来证明对 $n=1,2,3,\cdots$ 来说 $P(n)$ 为真。即必须证明 $P(1)$ 为真，以及对 $n=1,2,3,\cdots$ 来说蕴涵式 $P(n)$ 蕴涵 $P(n+1)$ 为真。

基础步骤：$P(0)$ 为真，因为 $1=1(1+1)/2$。

归纳步骤：假定 $P(n)$ 成立，使得

$$1+2+\cdots+n=n(n+1)/2$$

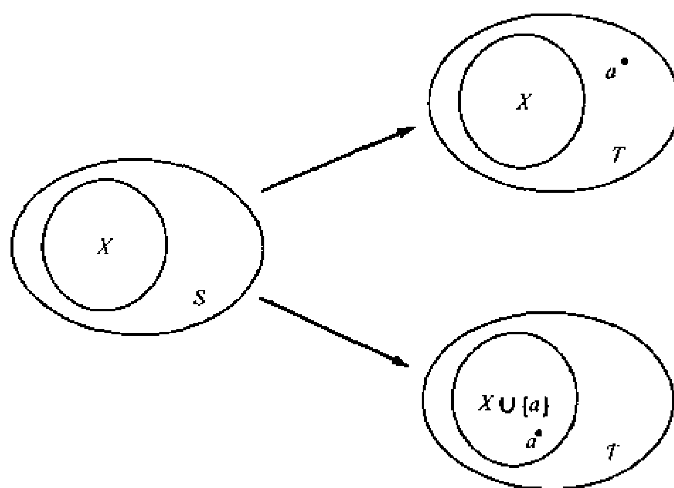


图 3-4 生成有 $n+1$ 个元素的集合的子集，这里 $T=S\cup\{a\}$

在这个假定之下，必须证明 $P(n+1)$ 为真，即

$$1+2+\cdots+n+n+1=(n+1)[(n+1)+1]/2=(n+1)(n+2)/2$$

也为真。对 $P(n)$ 里的等式的两边加上 $n+1$ 就得到

$$\begin{aligned}1+2+\cdots+n+(n+1)&=n(n+1)/2+(n+1)\\&=[(n/2)+1](n+1)\\&=(n+1)(n+2)/2\end{aligned}$$

最后这个等式证明 $P(n+1)$ 为真。这样就完成了归纳步骤，证毕。 ■

例 10 用数学归纳法证明：对每个满足 $n\geqslant 4$ 的正整数 n 来说有 $2^n<n!$。

解 设 $P(n)$ 是命题：$2^n<n!$。

基础步骤：为了对 $n\geqslant 4$ 来说证明这个不等式，就需要让基础步骤是 $P(4)$。注意 $P(4)$ 为真，因为 $2^4=16<4!=24$。

归纳步骤：假定 $P(n)$ 为真，即假定 $2n<n!$。必须证明 $P(n+1)$ 为真。即必须证明

$2^{n+1}<(n+1)!$。对不等式 $2^n<n!$ 的两边乘上 2 就得到

$$
\begin{aligned}
2\cdot 2^n &< 2\cdot n! \\
&< (n+1)\cdot n! \\
&= (n+1)!
\end{aligned}
$$

这说明当 $P(n)$ 为真时 $P(n+1)$ 为真。这样就完成了证明的归纳步骤。因此，就得出对每个满足 $n\geqslant 4$ 的正整数 n 来说 $2^n<n!$ 为真。■

例 11 用数学归纳法证明对德摩根律之一的下述推广：每当 $A_1, A_2, \cdots, A_n$ 都是全集 U 的子集并且 $n\geqslant 2$ 时就有

$$\overline{\bigcap_{k=1}^{n} A_k}=\bigcup_{k=1}^{n}\overline{A_k}$$

解 设 $P(n)$ 是对 n 个集合来说的上述恒等式。

基础步骤：命题 $P(2)$ 断言 $\overline{A_1\cap A_2}=\overline{A_1}\cup\overline{A_2}$。这是德摩根律之一；在第 1.5 节里证明过它。

归纳步骤：假定 $P(n)$ 为真，即每当 $A_1, A_2, \cdots, A_n$ 都是全集 U 的子集并且 $n\geqslant 2$ 时，就有

$$\overline{\bigcap_{k=1}^{n} A_k}=\bigcup_{k=1}^{n}\overline{A_k}$$

为了完成归纳步骤，必须证明若对 U 的任意 n 个子集来说这个等式成立，则对 U 的任意 $n+1$ 个子集来说它也必然是有效的。假定 $A_1, A_2, \cdots, A_n, A_{n+1}$ 都是 U 的子集。当假定归纳假设成立时，就得出

$$
\begin{aligned}
\overline{\bigcap_{k=1}^{n+1} A_k} &= \overline{\bigcap_{k=1}^{n} A_k\cap A_{n+1}} \\
&= \overline{\bigcap_{k=1}^{n} A_k}\cup\overline{A_{n+1}} && \text{（根据德摩根定律）} \\
&= \bigcup_{k=1}^{n}\overline{A_k}\cup\overline{A_{n+1}} && \text{（根据归纳假设）} \\
&= \bigcup_{k=1}^{n+1}\overline{A_k}
\end{aligned}
$$

这样就完成了归纳法证明。■

下一个例子说明如何用数学归纳法证明关于用形如字母“L”的碎片覆盖棋盘的结果。

例 12 设 n 是正整数。证明：可以用 L 形状的碎片来铺满去掉 1 个格子的任何 $2^n\times 2^n$ 棋盘，其中这些碎片一次覆盖 3 个格子，如图 3-5 所示。



图 3-5 一个 L 形状的碎片

解 设 $P(n)$ 是命题：可以用 L 形状的碎片来铺满去掉 1 个格子的任何 $2^n\times 2^n$ 棋盘。可以用数学归纳法证明对所有正整数 n 来说 $P(n)$ 为真。

基础步骤：命题 $P(1)$ 为真，因为可以用 1 个 L 形状的碎片来

铺满去掉1个格子的4个2×2棋盘，如图3-6所示。



图3-6 铺满去掉1个格子的2×2棋盘

归纳步骤：假定$P(n)$为真；即假定可以用L形状的碎片来铺满去掉1个格子的任何$2^n \times 2^n$棋盘。必须证明在这个假定下$P(n+1)$也必然为真；即可以用L形状的碎片来铺满去掉1个格子的任何$2^{n+1} \times 2^{n+1}$棋盘。

为了看出这一点，考虑去掉1个格子的$2^{n+1} \times 2^{n+1}$棋盘。通过在两个方向上都一分为二，把这个棋盘分成$2^n \times 2^n$大小的4个棋盘。在图3-7里对此进行说明。这4个棋盘中的3个里没有去掉任何格子。第4个$2^n \times 2^n$棋盘去掉了1个格子，所以根据归纳假设，可以用L形状的碎片来覆盖它。现在暂时去掉另外3个$2^n \times 2^n$棋盘的一角上的格子，这些角是原来的较大的棋盘的中心，如图3-8所示。根据归纳假设，可以用L形状的碎片来铺满这3个去掉了1个格子的$2^n \times 2^n$棋盘中的每一个。另外，可以用1个L形状的碎片来铺满这3个暂时去掉的格子。因此，可以用L形状的碎片来铺满整个的$2^{n+1} \times 2^{n+1}$棋盘。证毕。 ■

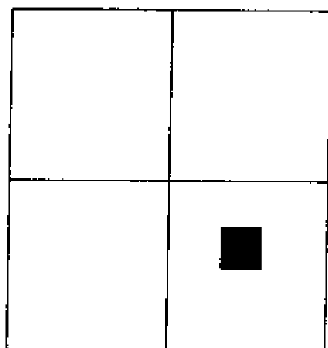


图3-7 把一个$2^{n+1} \times 2^{n+1}$棋盘分成四个$2^n \times 2^n$棋盘

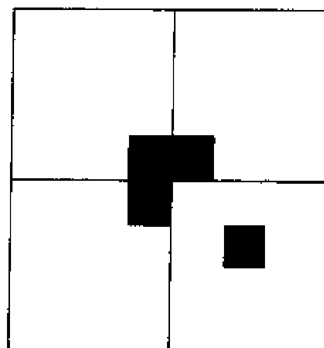


图3-8 铺满去掉了1个格子的$2^{n+1} \times 2^{n+1}$棋盘

3.2.5 数学归纳法的第二原理

存在另一种形式的数学归纳法，它的证明里常常是很有用的。在这种形式里，使用与前面同样的基础步骤，但是使用不同的归纳步骤。假定对$k=1, \cdots, n$来说$P(k)$为真，证明在这个假定的基础之上$P(n+1)$也必然为真。这称为*数学归纳法的第二原理*。总结一下用来证明对所有正整数n来说$P(n)$为真的两个步骤：

1. *基础步骤*。证明命题$P(1)$为真。

2. *归纳步骤*。证明对每个正整数n来说$[P(1) \wedge P(2) \wedge \cdots \wedge P(n)] \rightarrow P(n+1)$为真。

数学归纳法的这两种形式是等价的；即假定其中一种是有效的证明技术，可以证明另外一种也是。把这一点留作习题让读者去证明它。现在给出一个例子说明如何使用数学归纳法

的第二原理。

例 13 证明：若 n 是大于 1 的整数，则 n 可以写成素数之积。

解 设 $P(n)$ 是命题：n 可以写成素数之积。

基础步骤：$P(2)$ 为真，因为 2 可以写成一个素数之积，即它自身。（注意 $P(2)$ 是需要证明的第一个情形。）

归纳步骤：假定对所有满足 $k \leqslant n$ 的正整数 k 来说 $P(k)$ 为真。要完成归纳步骤，就必须证明在这个假定下 $P(n+1)$ 为真。

有两种要考虑的情形，即当 $n+1$ 是素数时和当 $n+1$ 是合数时。若 $n+1$ 是素数，则立即看出 $P(n+1)$ 为真。否则，$n+1$ 是合数并且可以写成满足 $2 \leqslant a \leqslant b \leqslant n+1$ 的两个整数 a 和 b 之积。根据归纳假设，a 和 b 都可以写成素数之积。因此，若 $n+1$ 是合数，则它可以写成素数之积，即在 A 的因子分解中的那些素数与在 b 的因子分解中的那些素数之积。 ■

注意 因为 1 是素数之积，即不包含任何素数的空积，所以可以在例 13 里用 $P(1)$ 作为基础步骤来开始证明。没有选择这样做是因为许多人对此感到迷惑不解。

注意例 13 完成了对算术基本定理的证明，该定理断言：每个非负整数可以唯一地写成以非减顺序排列的素数之积。在第 2.5 节里证明过整数最多有一种这样的素因子分解。例 13 证明至少有一种这样的分解。

用数学归纳法原理来代替数学归纳法的第二原理去证明例 13 里的结果，这是困难的。不过，例 14 说明用数学归纳法原理或数学归纳法第二原理都容易证明某些结果。

例 14 证明：可以仅用 4 分和 5 分邮票来组成等于或超过 12 分的每种邮资。

解 将要用数学归纳法原理来证明这个结果。然后给出用数学归纳法第二原理的证明。设 $P(n)$ 是命题：可以用 4 分和 5 分邮票来组成 n 分邮资。

首先使用数学归纳法原理。

基础步骤：可以用 3 个 4 分邮票来组成 12 分邮资。

归纳步骤：假定 $P(n)$ 为真，所以可以用 4 分和 5 分邮票来组成 n 分邮资。若至少用了一个 4 分邮票，则用一个 5 分邮票代替它，就组成 $n+1$ 分邮资。若没有用任何 4 分邮票，则仅用了 5 分的邮票来组成 n 分邮资。因为 $n \geqslant 12$，所以至少用了 3 个 5 分邮票。所以，用 4 个 4 分邮票来代替 3 个 5 分邮票，就组成了 $n+1$ 分邮资。这完成了归纳步骤以及根据数学归纳法原理的证明。

其次，将要使用数学归纳法的第二原理。将要证明可以组成 12，13，14 和 15 分邮资，然后证明如何对 $n \geqslant 15$ 来说从 $n-3$ 分邮资得出 $n+1$ 分邮资。

基础步骤：可以分别用 3 个 4 分邮票，2 个 4 分邮票和 1 个 5 分邮票，1 个 4 分邮票和 2 个 5 分邮票，以及 3 个 5 分邮票，来组成 12，13，14 和 15 分邮资。

归纳步骤：设 $n \geqslant 15$。假定可以组成 k 分邮资，其中 $12 \leqslant k \leqslant n$。为了组成 $n+1$ 分邮资，用组成 $n-3$ 分邮资的邮票加上 1 个 4 分邮票。这完成了归纳步骤以及根据数学归纳法第二原理的证明。

（除了这里描述的方法以外，还有处理这个问题的其他方法。读者能否找出不使用数学归纳法的解答？） ■

注意 例 14 说明如何让数学归纳法第二原理适应于处理某些情形，其中仅对充分大的 n 值来说归纳步骤才是有效的。具体说来，为了证明对 $n=k, k+1, k+2, \cdots$ 来说 $P(n)$ 为真，其中 k 是整数，首先证明 $P(k), P(k+1), P(k+2), \cdots, P(l)$ 都为真（基础步骤），然后证明对每个整数 $n \geq l$ 来说 $[P(k) \wedge P(k+1) \wedge P(k+2) \wedge \cdots \wedge P(n)] \rightarrow P(n+1)$ 为真（归纳步骤）。例如，例 14 解答里的第二个证明的基础步骤证明 $P(12), P(13), P(14)$ 和 $P(15)$ 都为真。需要分别地证明这些情形，因为归纳步骤证明 $[P(12) \wedge P(13) \wedge \cdots \wedge P(n)] \rightarrow P(n+1)$，它仅当 $n \geq 15$ 时才成立。

在下面几节里将要讨论数学归纳法的两个重要应用。第一个应用涉及到定义序列而不给出明确的项公式。第二个应用涉及到证明计算机程序是正确的。

练习

1. 找出前 n 个正偶数之和的公式。
2. 用数学归纳法证明在练习题里找出的公式。
3. 用数学归纳法证明：每当 n 是非负整数时就有
   $$3+3\cdot5+3\cdot5^2+\cdots+3\cdot5^n=3(5^{n+1}-1)/4$$
4. 用数学归纳法证明：每当 n 是非负整数时就有
   $$2-2\cdot7+2\cdot7^2-\cdots+2(-7)^n=(1-(-7)^{n+1})/4$$
5. 通过对小的 n 值检查表达式
   $$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\cdots+\frac{1}{2^n}$$
   的值，来找出这个表达式的公式。用数学归纳法证明你的结果。
6. 通过对小的 n 值检查表达式
   $$\frac{1}{1\cdot2}+\frac{1}{2\cdot3}+\cdots+\frac{1}{n(n+1)}$$
   的值，来找出这个表达式的公式。用数学归纳法证明你的结果。
7. 证明：每当 n 是正整数时，就有 $1^2+2^2+\cdots+n^2=n(n+1)(2n+1)/6$。
8. 证明：每当 n 是正整数时，就有 $1^3+2^3+\cdots+n^3=[n(n+1)/2]^2$。
9. 证明：每当 n 是非负整数时，就有 $1^2+3^2+5^2+\cdots+(2n+1)^2=(n+1)(2n+1)(2n+3)/6$。
10. 证明：每当 n 是正整数时，就有 $1\cdot1!+2\cdot2!+\cdots+n\cdot n!=(n+1)!-1$。

*11. 用数学归纳法证明：若 $h>-1$，则对所有非负整数 n 来说 $1+nh \leq (1+h)^n$。这称为伯努利不等式。

12. 证明：每当 n 是大于 6 的正整数时，就有 $3^n<n!$。
13. 证明：每当 n 是大于 4 的正整数时，就有 $2^n>n^2$。
14. 用数学归纳法证明：每当 n 是大于 1 的正整数时，就有 $n!<n^n$。
15. 用数学归纳法证明：每当 n 是正整数时，就有
    $$1\cdot2+2\cdot3+\cdots+n(n+1)=n(n+1)(n+2)/3$$

16. 用数学归纳法证明：$1\cdot2\cdot3+2\cdot3\cdot4+\cdots+n(n+1)(n+2)=n(n+1)(n+2)(n+3)/4$。
17. 证明：每当 n 是正整数时，就有$1^2-2^2+3^2-\cdots+(-1)^{n-1}n^2=(-1)^{n-1}n(n+1)/2$。
18. 证明：每当 n 是大于 1 的正整数时，就有

$$1+\frac{1}{4}+\frac{1}{9}+\cdots+\frac{1}{n^2}<2-\frac{1}{n}$$

19. 证明：可以仅用 3 分邮票和 5 分邮票，来组成大于 7 分的正整数分的任何邮资。
20. 用数学归纳法证明：每当 n 是非负整数时，就有 3 整除 n^3+2n。
21. 用数学归纳法证明：每当 n 是非负整数时，就有 5 整除 n^5-n。
22. 用数学归纳法证明：每当 n 是非负整数时，就有 6 整除 n^3-n。
*23. 用数学归纳法证明：每当 n 是正奇数时，就有 n^2-1 被 8 整除。
24. 用数学归纳法证明：每当 n 是大于 3 的整数时，就有 $n^2-7n+12$ 是非负的。
25. 用数学归纳法证明：每当 n 是大于或等于 2 的整数时，一个带有 n 个元素的集合就有$n(n-1)/2$ 个恰好包含两个元素的子集合。
*26. 用数学归纳法证明：每当 n 是大于或等于 3 的整数时，一个带有 n 个元素的集合，就有$n(n-1)(n-2)/6$ 个恰好包含三个元素的子集合。
27. 用数学归纳法证明：每当 n 是正整数时，就有$\sum_{j=1}^{n} j^4=n(n+1)(2n+1)(3n^2+3n-1)/30$。
28. 对哪些非负整数 n 来说有 $n^2\leqslant n!$？用数学归纳法证明你的答案。
29. 对哪些非负整数 n 来说有 $2n+3\leqslant 2^n$？用数学归纳法证明你的答案。
30. 用数学归纳法证明：每当 n 是正整数时，就有$1/(2n)\leqslant[1\cdot3\cdot5\cdot\cdots\cdot(2n-1)]/(2\cdot4\cdot\cdots\cdot2n)$。
31. a）确定仅用 5 分邮票和 6 分邮票可以组成哪些数量的邮资。
    b）用数学归纳法证明你对 a）的答案。
    c）用数学归纳法第二原理证明你对 a）的答案。
32. 仅用 10 分硬币和 25 分硬币可以组成哪些数量的钱币？
33. 一种自动柜员机只有 20 美元和 50 美元。假定这种机器有这两种美元的无限的供应量，这种机器可以分发哪些数量的美元？用数学归纳法的一种形式证明你的答案。
34. 用数学归纳法证明：$\sum_{k=1}^{n} k2^k=(n-1)2^{n+1}+2$。
35. 证明：$\sum_{\{a_1,\cdots,a_k\}\subseteq\{1,2,\cdots,n\}}\frac{1}{a_1a_2\cdots a_k}=n$。（在这里求和是在 n 个最小正整数的集合的所有非空子集上进行。）
36. 用数学归纳法证明：给定有 $n+1$ 个正整数的集合，这些正整数都不超过 $2n$，在这个集合里至少存在一个整数，它整除在这个集合里的另外一个整数。
37. （本题需要微积分知识）用数学归纳法证明：每当 n 是正整数时，就有 $f(x)=x^n$ 的导数等于 nx^{n-1}。（对归纳步骤来说，使用导数的乘法规则。）
38. 假定

$$\mathbf{A}=\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$$

其中 a 和 b 都是正实数。证明：对每个正整数 n 来说

$$\mathbf{A}^n=\begin{bmatrix} a^n & 0 \\ 0 & b^n \end{bmatrix}$$

39. 假定 **A** 和 **B** 都是方阵且具有性质 $\mathbf{AB}=\mathbf{BA}$。证明：对每个正整数 n 来说有 $\mathbf{AB}^n=\mathbf{B}^n\mathbf{A}$。
40. 假定 m 是正整数。用数学归纳法证明：若 a 和 b 都是整数且 $a\equiv b\pmod m$，则每当 k 是非负整数时，就有 $a^k\equiv b^k\pmod m$。
41. 用数学归纳法证明：若 A_1，A_2，…，A_n 和 B 都是集合，则

$$(A_1\cup A_2\cup\cdots\cup A_n)\cap B=(A_1\cap B)\cup(A_2\cap B)\cup\cdots\cup(A_n\cap B)$$

42. 证明：若 A_1，A_2，…，A_n 和 B_1，B_2，…，B_n 都是集合，且对 $k=1$，2，…，n 来说 $A_k\subseteq B_k$，则

a) $\bigcup_{k=1}^{n} A_k\subseteq\bigcup_{k=1}^{n} B_k$　　b) $\bigcap_{k=1}^{n} A_k\subseteq\bigcap_{k=1}^{n} B_k$

43. 用数学归纳法证明：若 A_1，A_2，…，A_n 都是论域 U 的子集，则

$$\overline{\bigcup_{k=1}^{n} A_k}=\bigcap_{k=1}^{n}\overline{A_k}$$

44. 用数学归纳法证明：每当 p_1，p_2，…，p_n 都是命题时，$\neg(p_1\vee p_2\vee\cdots\vee p_n)$ 就等价于 $\neg p_1\wedge\neg p_2\wedge\cdots\wedge\neg p_n$。

*45. 证明：每当 p_1，p_2，…，p_n 都是命题时，

$$[(p_1\to p_2)\wedge(p_2\to p_3)\wedge\cdots\wedge(p_{n-1}\to p_n)]\to[(p_1\wedge p_2\wedge\cdots\wedge p_{n-1})\to p_n]$$

就是重言式。

46. 用几何级数各项之和的公式来对下面的和求值。
    a) $4+4\cdot 3+4\cdot 3^2+\cdots+4\cdot 3^8$
    b) $3+3\cdot 2^2+3\cdot 2^4+\cdots+3\cdot 2^{10}$
    c) $1-2+2^2-2^3+\cdots+(-1)^n 2^n$
47. 下述的关于所有马都是相同颜色的“证明”的错误是什么？

设 $P(n)$ 是命题：在 n 匹马的集合里所有马都是相同颜色的。显然，$P(1)$ 为真。现在假定 $P(n)$ 为真，所以在任意 n 匹马的集合里所有马都是相同颜色的。考虑任意的 $n+1$ 匹马；把这些马编号成 1，2，3，…，n，$n+1$。现在这些马中的前 n 匹必然都有相同颜色，而且这些马中的后 n 匹也必然都有相同颜色。因为前 n 匹马的集合与后 n 匹马的集合重叠，所以所有 $n+1$ 匹马必然都是相同颜色。这就证明了 $P(n+1)$ 为真并且完成了归纳证明。

*48. 找出下列证明的错误：每当 a 是非零实数时，就对所有非负整数 n 来说有 $a^n=1$。

基础步骤：根据 a^0 的定义，$a^0=1$ 为真。

归纳步骤：假定对满足 $k\leqslant n$ 的所有非负整数 k 来说 $a^k=1$。然后注意

$$a^{n+1}=\frac{a^n\cdot a^n}{a^{n-1}}=\frac{1\cdot 1}{1}=1.$$

*49. 通过证明从良序性得出数学归纳法第二形式，来证明数学归纳法第二形式是有效的证明方法。

*50. 证明：为了证明对所有正整数 n 来说 $P(n)$ 为真，数学归纳法的下列形式是有效的证明方法。

基础步骤：$P(1)$ 和 $P(2)$ 都为真。

归纳步骤：对每个正整数 n 来说，若 $P(n)$ 和 $P(n+1)$ 都为真，则 $P(n+2)$ 为真。

在练习 51 和练习 52 中，H_n 表示第 n 个调和数。

*51. 用数学归纳法证明：每当 n 是非负整数时，就有 $H_{2^n}\leqslant 1+n$。

*52. 用数学归纳法证明：$H_1+H_2+\cdots+H_n=(n+1)H_n-n$。

*53. 证明：$1+\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{3}}+\cdots+\frac{1}{\sqrt{n}}>2(\sqrt{n+1}-1)$。

*54. 证明：若 n 条直线中任何两条都不平行，任何三条都不共点，则这些直线把平面分割成 $(n^2+n+2)/2$ 个区域。

**55. 设 a_1，a_2，…，a_n 都是正实数。这些数的算术平均值定义成 $A=(a_1+a_2+\cdots+a_n)/n$，而这些数的几何平均值定义成 $G=(a_1a_2\cdots a_n)^{1/n}$。用数学归纳法证明：$A\geqslant G$。

*56. 用数学归纳法证明：每当 n 是正整数时，就有 21 整除 $4^{n+1}+5^{2n-1}$。

57. 用数学归纳法证明第 2.5 节引理 2，该引理说：若 p 是素数且 $p\mid a_1a_2\cdots a_n$，其中对 $i=1,2,\cdots,n$ 来说 a_i 是整数，则对某个整数 i 来说 $p\mid a_i$。

*58. 良序性可以被用来证明：两个正整数有唯一的最大公因子。设 a 和 b 都是正整数，设 S 是形如 $as+bt$ 的正整数的集合，其中 s 和 t 都是正整数。

a) 证明：S 非空。

b) 用良序性证明：S 有最小元 c。

c) 证明：若 d 是 a 和 b 的公因子，则 d 是 c 的因子。

d) 证明：$c\mid a$ 和 $c\mid b$。[提示：首先假定 $c\nmid a$。则 $a=qc+r$，其中 $0<r<c$。证明 $r\in S$，这与对 c 的选择相矛盾。]

e) 从 (c) 和 (d) 得出：a 和 b 的最大公因子存在。通过证明两个正整数的最大公因子是唯一的，来完成证明。

*59. 证明：若 a_1，a_2，…，a_n 是 n 个不同的实数，则无论在它们的乘积中插入多少对括号，计算这 n 个数之积都要使用 $n-1$ 次乘法。[提示：利用数学归纳法第二原理并且考虑最后一次的乘法。]

60. 用 L 形状的碎片去构造去掉左上角格子的 4×4 棋盘的铺砖图案。

61. 用 L 形状的碎片去构造去掉左上角格子的 8×8 棋盘的铺砖图案。

62. 证明或反驳：每当 n 是正整数时，就可以用 L 形状的碎片完全地覆盖下面形状的所有棋盘。

a) 3×2^n　　b) 6×2^n　　c) $3^n\times3^n$　　d) $6^n\times6^n$

*63. 证明：用去掉了一个 $1\times1\times1$ 立方体的 $2\times2\times2$ 立方体，可以完全地覆盖去掉了一个 $1\times1\times1$ 立方体的三维 $2^n\times2^n\times2^n$ 棋盘。

*64. 证明：若 $n>5$，n 是奇数，且 n 不能被 3 整除，则用 L 形状的碎片可以完全地覆盖去掉了一个格子的 $n\times n$ 棋盘。

65. 证明：可以用 L 形状的碎片来铺满去掉了一个角上格子的 5×5 棋盘。

*66. 找出不能用 L 形状的碎片来铺满的去掉了一个格子的 5×5 棋盘。证明对这个棋盘来说这样的铺砖图案不存在。

67. 设 a 是整数而 d 是正整数。证明：满足 $a=dq+r$ 和 $0\leqslant r<d$ 的整数 q 和 r 是唯一的。在例 1 里证明过它们存在。

☞68. 用数学归纳法原理证明：若 $P(k)$ 为真，并且对满足 $n\geqslant k$ 的所有正整数 n 来说，蕴涵式 $P(n)\rightarrow P(n+1)$ 为真，则对 $n=k,k+1,k+2,\cdots$ 来说 $P(n)$ 为真，其中 k 是整数。

**69. 你能用良序性证明下面的命题吗：“每一个正整数都可以用不超过 15 个英语单词来描述”？

3.3 递归定义

3.3.1 引言

有时难以用明确的方式来定义一个对象。不过，用这个对象来定义它自身，这也许是容易的。这种过程称为递归。例如，图 3-9 里所示的图画是递归地产生的。首先，给出一幅原来的图画。然后完成一个过程，它在前一幅图画的中央递归地放上更小的图画。

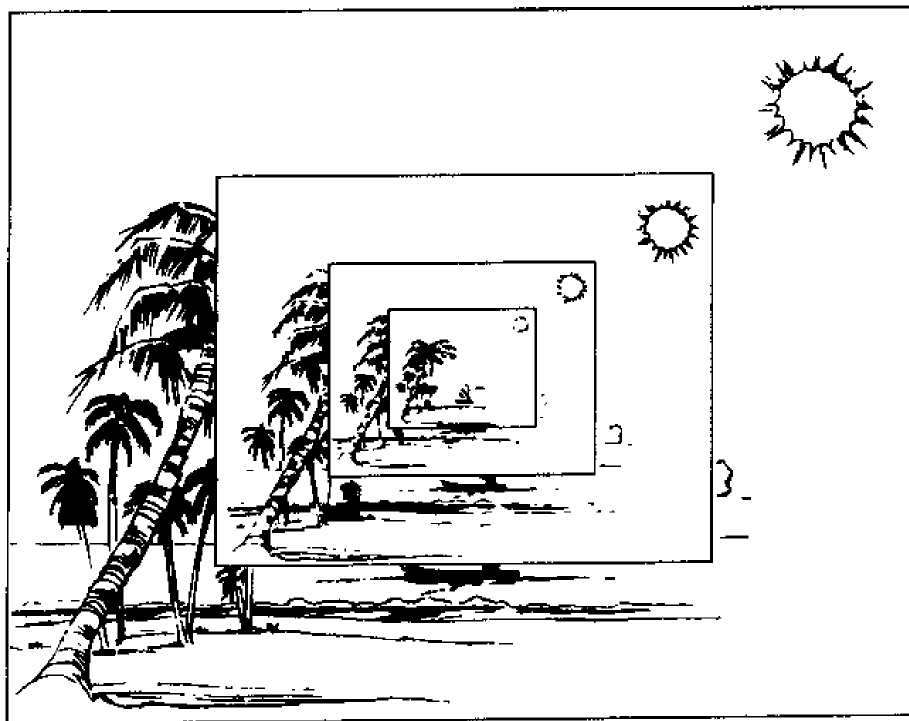


图 3-9 递归地定义的图画

可以用递归来定义序列、函数和集合。在前面的讨论里用显式的公式来规定序列里的

项。例如，对 $n=0, 1, 2, \cdots$ 来说用 $a_n=2^n$ 来给出 2 的幂的序列。不过，通过给出这个序列的第一项，即 $a_0=1$，以及从该序列前面一项来求当前项的规则，即对 $n=0, 1, 2, \cdots$ 来说 $a_{n+1}=2a_n$，也可以定义这个序列。

3.3.2 递归地定义函数

为了定义以非负整数集合作为其定义域的函数，就要

1. 规定这个函数在 0 处的值。
2. 给出从较小的整数处的值来求出当前的值的规则。

这样的定义称为*递归定义*或*归纳定义*。

例 1 假定 f 是用

$$f(0)=3$$
$$f(n+1)=2f(n)+3$$

来递归地定义的。求出 $f(1), f(2), f(3)$ 和 $f(4)$。

解 从这个递归定义得出

$$f(1)=2f(0)+3=2\cdot3+3=9$$
$$f(2)=2f(1)+3=2\cdot9+3=21$$
$$f(3)=2f(2)+3=2\cdot21+3=45$$
$$f(4)=2f(3)+3=2\cdot45+3=93$$

■

许多函数都可以利用它们的递归定义来研究。阶乘函数就是一个这样的例子。

例 2 给出阶乘函数 $F(n)=n!$ 的归纳定义。

解 可以通过规定阶乘函数的初值，即 $F(0)=1$，并且给出从 $F(n)$ 求出 $F(n+1)$ 的规则，来定义这个函数。要得出这个结果，注意通过乘以 $n+1$ 就从 $n!$ 计算出 $(n+1)!$。因此，所需要的规则是 $F(n+1)=(n+1)F(n)$。 ■

为了从在例 2 里求出的递归定义来确定阶乘函数的一个值，比如 $F(5)=5!$，有必要多次使用说明如何用 $F(n)$ 表示 $F(n+1)$ 的规则：

$$F(5)=5F(4)=5\cdot4F(3)=5\cdot4\cdot3F(2)=5\cdot4\cdot3\cdot2F(1)$$
$$=5\cdot4\cdot3\cdot2\cdot1\cdot F(0)=5\cdot4\cdot3\cdot2\cdot1\cdot1=120$$

一旦 $F(0)$ 是出现的唯一的函数值，就不需要任何更多的归约。剩下来要做的唯一事情是把 $F(0)$ 的值插入到公式里。

递归地定义的函数是严格定义的。这是数学归纳法原理的一个后果。（见本节末的练习 44。）在下面的例子里给出递归定义的其他例子。

例 3 给出 a^n 的递归定义，其中 a 是非零实数而且 n 是非负整数。

解 这个递归定义包括两个部分。首先规定 a^0，即 $a^0=1$。然后给出从 a^n 求出 a^{n+1} 的规则，即对 $n=0, 1, 2, 3, \cdots$ 来说 $a^{n+1}=a\cdot a^n$。这两个等式对所有非负整数唯一地定义了 a^n。 ■

例 4 给出

$$\sum_{k=0}^{n} a_k$$

的递归定义。

解 这个递归定义的第一部分是

$$\sum_{k=0}^{0} a_k = a_0$$

第二部分是

$$\sum_{k=0}^{n+1} a_k = \sum_{k=0}^{n} a_k + a_{n+1}$$ ■

在函数的某些递归定义里，规定了函数在前 k 个正整数处的值，而且给出了从一个较大的整数之前的部分或全部 k 个整数处的函数值来确定在该整数处的函数值的规则。从数学归纳法第二原理可以得出结论说这样的定义产生严格定义的函数。（见本节末的练习 45。）

例 5 斐波那契[○]数 f_0，f_1，f_2，…是用等式 $f_0=0$，$f_1=1$，以及对 $n=2, 3, 4, \cdots$来说

$$f_n = f_{n-1} + f_{n-2}$$

来定义的。斐波那契数 f_2，f_3，f_4，f_5，f_6 是什么？

解 因为这个定义的第一部分说 $f_0=0$ 和 $f_1=1$，所以从这个定义的第二部分得出

$$f_2 = f_1 + f_0 = 1+0 = 1$$
$$f_3 = f_2 + f_1 = 1+1 = 2$$
$$f_4 = f_3 + f_2 = 2+1 = 3$$
$$f_5 = f_4 + f_3 = 3+2 = 5$$
$$f_6 = f_5 + f_4 = 5+3 = 8$$ ■

可以用斐波那契数的递归定义来证明这些数的许多性质。在下一个例子里给出一个这样的性质。

例 6 证明：每当 $n \geqslant 3$ 时就有 $f_n > \alpha^{n-2}$，其中 $\alpha = (1+\sqrt{5})/2$。

解 可以用数学归纳法第二原理来证明这个不等式。设 $P(n)$ 是命题：$f_n > \alpha^{n-2}$。想要证明每当 n 是大于或等于 3 的整数时就有 $P(n)$ 为真。

首先，注意到

$$\alpha < 2 = f_3, \alpha^2 = (3+\sqrt{5})/2 < 3 = f_4$$

○ 斐波那契（Fibonacci，约 1180—1228，filius Bonacci 或 "Bonacci 之子" 的简称） 斐波那契也被称为比萨的列奥那多（Leonardo）。他诞生在意大利的商业中心比萨。斐波那契是一位商人，他遍游中东各地，在那里接触到阿拉伯数学。在他的著作《算书》（*Liber Abaci*）中，斐波那契向欧洲人介绍了阿拉伯的数字记号和算术的算法。正是在这本书里出现了他的著名的兔子问题（在第 5.1 节里描述）。斐波那契还写过关于几何学和三角学以及关于丢番图方程的各种书，丢番图方程是关于寻找方程的整数解的。

所以$P(3)$和$P(4)$都为真。现在假定$P(k)$为真，即对所有满足$3\leqslant k\leqslant n$的整数k来说有$f_k>\alpha^{k-2}$，其中$n\geqslant 4$。必须证明$P(n+1)$为真，即$f_{n+1}>\alpha^{n-1}$。因为α是$x^2-x-1=0$的解（二次方程求根公式说明这一点），所以得出$\alpha^2=\alpha+1$。因此，

$$\alpha^{n-1}=\alpha^2\cdot\alpha^{n-3}=(\alpha+1)\alpha^{n-3}=\alpha\cdot\alpha^{n-3}+1\cdot\alpha^{n-3}=\alpha^{n-2}+\alpha^{n-3}$$

根据归纳假设，若$n\geqslant 5$，则得出

$$f_{n-1}>\alpha^{n-3}，\ f_n\alpha^{n-2}$$

因此就有

$$f_{n+1}=f_n+f_{n-1}>\alpha^{n-2}+\alpha^{n-3}=\alpha^{n-1}$$

由此得出$P(n+1)$为真，证毕。 ■

注意 归纳步骤证明了每当$n\geqslant 4$时，从对$3\leqslant k\leqslant n$来说$P(k)$为真的假定就得出$P(n+1)$。因此，归纳步骤没有证明$P(3)\rightarrow P(4)$。所以，不得不单独证明$P(4)$为真。

现在可以证明：欧几里德算法用$O(\log b)$次除法来求出正整数a和b的最大公因子，其中$a\geqslant b$。

定理1 拉梅[⊖]定理 设a和b是满足$a\geqslant b$的正整数。则欧几里德算法为了求出$\gcd(a,b)$而使用的除法的次数小于或等于b的十进制位数的5倍。

证 回忆一下，当用欧几里德算法来求出满足$a\geqslant b$的$\gcd(a,b)$时，得出了下面的等式序列（其中$a=r_0$而$b=r_1$）。

$$\begin{aligned}
r_0 &= r_1q_1+r_2,\quad 0\leqslant r_2<r_1\\
r_1 &= r_2q_2+r_3,\quad 0\leqslant r_3<r_2\\
&\ \ \vdots\\
r_{n-2} &= r_{n-1}q_{n-1}+r_n,\quad 0\leqslant r_n<r_{n-1}\\
r_{n-1} &= r_nq_n
\end{aligned}$$

在这里为了求出$\gcd(a,b)$而使用了n次除法。注意商q_1，q_2，…，q_{n-1}都至少是1。另外，$q_n\geqslant 2$，因为$r_n<r_{n-1}$。这就蕴涵着

⊖ 加布里尔·拉梅（Gabriel Lame，1795—1870） 拉梅于1813年进入工业高等专科学校，1817年毕业。他在米内兹高等专科学校继续接受教育，于1820年毕业。

在1820年拉梅来到俄国，在那里被任命为圣彼得堡的公路与运输学校的校长。在俄国期间他不仅教书，而且还设计道路和桥梁。他在1832年回到巴黎，在那儿帮助成立一家工程公司。不过，他很快就离开这个公司，接受工业高等专科学校的物理学教授职务，他担任这个职务直到1844年。担任这个职务期间，他作为工程顾问而活跃在与学术无关的事情上，担任煤矿的首席工程师并且参与铁路建设。

拉梅对数论、应用数学以及热力学都作出了开创性工作。他最著名的工作包括引入曲线坐标。他在数论上的工作包括对$n=7$证明费尔马最后定理，以及证明本文里给出的欧几里德算法所用除法次数的上界。

在所有时代都是最重要的数学家之一高斯看来，拉梅是当时最出色的法国数学家。不过，法国数学家认为他太实际化，而法国科学家则认为他太理论化。

$$r_n \geqslant 1 = f_2$$
$$r_{n-1} \geqslant 2r_n \geqslant 2f_2 = f_3$$
$$r_{n-2} \geqslant r_{n-1} + r_n \geqslant f_3 + f_2 = f_4$$
$$\vdots$$
$$r_2 \geqslant r_3 + r_4 \geqslant f_{n-1} + f_{n-2} = f_n$$
$$b = r_1 \geqslant r_2 + r_3 \geqslant f_n + f_{n-1} = f_{n+1}$$

由此得出，若欧几里德算法为了求出满足 $a \geqslant b$ 的 $\gcd(a,b)$ 而使用了 n 次除法，则 $b \geqslant f_{n+1}$。从例 6 中知道，对 $n>2$ 来说 $f_{n+1} > \alpha^{n-1}$，其中 $\alpha = (1+\sqrt{5})/2$。因此得出 $b > \alpha^{n-1}$。另外，因为 $\log_{10}\alpha \sim 0.208 > 1/5$，所以可以看出

$$\log_{10} b > (n-1)\log_{10}\alpha > (n-1)/5。$$

因此，$n-1 < 5 \cdot \log_{10} b$。现在假定 b 有 k 个十进数位。则 $b < 10^k$ 而且 $\log_{10} b < k$。由此得出 $n-1 < 5k$，而且因为 k 是整数，所以得出 $n \leqslant 5k$。证毕。 □

因为 b 的十进数位数等于 $\lfloor \log_{10} b \rfloor + 1$，它小于或等于 $\log_{10} b + 1$，故定理 1 说明求出满足 $a \geqslant b$ 的 $\gcd(a,b)$ 所需要的除法次数小于或等于 $5(\log_{10} b + 1)$。因为 $5(\log_{10} b + 1)$ 是 $O(\log b)$，故可看出每当 $a \geqslant b$ 时，欧几里德算法就用 $O(\log b)$ 次除法来求出 $\gcd(a,b)$。

3.3.3 递归地定义集合

递归定义常常用来定义集合。当这样做时，给出初始的一些元素。然后给出用来从已知属于集合的元素来构造集合的其他元素的规则。以这种方式描述的集合是严格定义的，用它们的递归定义可以证明关于它们的定理。下面是集合的递归定义的一些例子。

例 7 设 S 是用

$3 \in S$；

若 $x \in S$ 且 $y \in S$，则 $x + y \in S$

来递归地定义的。证明：S 是被 3 整除的正整数集合。（注意在这个定义里隐含着假定：所有属于 S 的东西都是用 S 的递归定义里的两个命题来生成的。）

解 设 A 是被 3 整除的所有正整数的集合。为了证明 $A = S$，必须证明 A 是 S 的子集而且 S 是 A 的子集。为了证明 A 是 S 的子集，必须证明被 3 整除的每个正整数都属于 S。将要用数学归纳法来证明它。

设 $P(n)$ 是命题：$3n$ 属于 S。基础步骤成立，因为根据 S 的递归定义的第一部分，$3 \times 1 = 3$ 是属于 S 的。为了证明归纳步骤，假定 $P(n)$ 为真，即 $3n$ 属于 S。因为 $3n$ 属于 S 而且因为 3 属于 S，所以从 S 的递归定义的第二部分得出 $3n + 3 = 3(n+1)$ 也属于 S。

为了证明 S 是 A 的子集，使用 S 的递归定义。首先，该定义的基础步骤规定 3 属于 S。因为 $3 = 3 \times 1$，所以所有在这个步骤里被规定属于 S 的元素都被 3 整除。为了完成证明，必须证明所有用该递归定义的第二部分所生成的属于 S 的元素都属于 A。这包括证明每当 x 和 y 都是 S 中的元素并且假定它们都属于 A 时，就有 $x + y$ 属于 A。现在若 x 和 y 都属于 A，则可以得出 $3 \mid x$ 和 $3 \mid y$。根据第 2.3 节的定理 1，得出 $3 \mid x + y$，证毕。 ■

在例 7 里集合的递归定义是典型的。首先，给出一组初始元素。其次，给出从已知属于集合的元素来生成新元素的规则。在定义里隐含着只有在初始元素中列出的元素，或者可以用构造新元素的规则来生成的那些元素才属于这个集合。

集合的递归定义的最普通的用途之一是定义各种系统里的合式公式。在下面的例子里说明这一点。

例 8 由变量、数字和 $\{+, -, *, /, \uparrow\}$ 中的运算符（其中 $*$ 代表乘法，$\uparrow$ 代表乘幂）所组成的合式公式定义成

若 x 是数字或变量，则 x 是合式公式；

若 f 和 g 是合式公式，则 $(f+g)$，$(f-g)$，$(f*g)$，(f/g) 和 $(f\uparrow g)$ 都是合式公式。

例如，根据这个定义，因为 x 和 3 都是合式公式，所以 $(x+3)$，$(x-3)$，$(x*3)$，$(x/3)$ 和 $(x\uparrow 3)$ 都是合式公式。接下来，因为 y 也是合式公式，所以 $((x+3)+y)(y-(x*3))$ 等也都是合式公式。(注意 $(3/0)$ 是合式公式，因为在这里只考虑语法。) ■

例 9 包含着 T、F、命题变元、以及运算符 $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ 的复合命题的合式公式定义成

T、F 和 p 都是合式公式，其中 p 是命题变元；若 p 和 q 是合式公式，则 $(\neg p)$，$(p \vee q)$，$(p \wedge q)$，$(p \rightarrow q)$，$(p \leftrightarrow q)$ 都是合式公式。

例如，若 p，q 和 r 是命题变元，则重复地使用这个递归定义，就证明 $(p \vee q)$，$(r \wedge \mathrm{T})$ 和 $(p \vee q) \rightarrow (r \wedge \mathrm{T})$ 都是合式公式。 ■

在对字符串的研究里常常使用递归定义。回忆一下第 1 章，字母表 Σ 上的字符串是 Σ 里符号的有穷序列。Σ 上的字符串的集合表示成 Σ^*。用连接运算可以组合两个字符串。字符串 x 和 y 的连接是字符串 x 后面接上字符串 y，表示成 xy。例如，$x = abra$ 和 $y = cadabra$ 的连接是 $xy = abracadabra$。当证明关于字符串的结果时，常常使用下面的递归定义。

例 10 *字符串集合的递归定义* 字母表 Σ 上的字符串的集合 Σ^* 递归地定义成：$\lambda \in \Sigma^*$，其中 λ 是不包含任何符号的空串，以及每当 $w \in \Sigma^*$ 和 $x \in \Sigma$ 时，就有 $wx \in \Sigma^*$。

这个定义的第一部分说明空串属于 Σ^*。第二部分说明把 Σ^* 的字符串与 Σ 的符号连接起来就产生新的字符串。 ■

字符串的长度是该字符串中符号的个数，它也可以递归地定义。

例 11 给出字符串 w 的长度 $l(w)$ 的递归定义。

解 字符串的长度可以定义成

$l(\lambda) = 0$；

$l(wx) = l(w) + 1$，若 $w \in \Sigma^*$ 而且 $x \in \Sigma$。 ■

下面的例子说明在证明里如何使用对字符串的递归定义。

例 12 用数学归纳法证明：$l(xy) = l(x) + l(y)$，其中 x 和 y 属于 Σ^*，即字母表 Σ 上的字符串的集合。

解 将要让证明是基于例10里给出的对集合Σ^*的递归定义。设$P(y)$是命题：每当$x \in \Sigma^*$时就有$l(xy)=l(x)+l(y)$。

基础步骤：为了完成基础步骤，必须证明$P(\lambda)$为真。即必须证明对所有$x \in \Sigma^*$来说有$l(x\lambda)=l(x)+l(\lambda)$。因为对每个字符串$x$来说$l(x\lambda)=l(x)=l(x)+0=l(x)+l(\lambda)$，所以$P(\lambda)$为真。

归纳步骤：为了完成归纳步骤，假定$P(y)$为真，而且证明这个假定蕴涵着每当$a \in \Sigma$时，就有$P(ya)$为真。需要证明的是，对每个$a \in \Sigma$来说有$l(xya)=l(x)+l(ya)$。为了证明这一点，注意根据$l(w)$的递归定义（在例11里给出），有$l(xya)=l(xy)+1$和$l(ya)=l(y)+1$。而且，根据归纳假设，有$l(xy)=l(x)+l(y)$。故得出$l(xya)=l(x)+l(y)+1=l(x)+l(ya)$。■

练习

1. 求出$f(1)$，$f(2)$，$f(3)$和$f(4)$，若$f(n)$递归地定义成$f(0)=1$，而且对$n=0,1,2,\cdots$来说
   a) $f(n+1)=f(n)+2$
   b) $f(n+1)=3f(n)$
   c) $f(n+1)=2^{f(n)}$
   d) $f(n+1)=f(n)^2+f(n)+1$
2. 求出$f(1)$，$f(2)$，$f(3)$，$f(4)$和$f(5)$，若$f(n)$递归地定义成$f(0)=3$，而且对$n=0,1,2,\cdots$来说
   a) $f(n+1)=-2f(n)$
   b) $f(n+1)=3f(n)+7$
   c) $f(n+1)=f(n)^2-2f(n)-2$
   d) $f(n+1)=3^{f(n)/3}$
3. 求出$f(2)$，$f(3)$，$f(4)$和$f(5)$。若f递归地定义成$f(0)=-1$，$f(1)=2$，而且对$n=1,2\cdots$来说
   a) $f(n+1)=f(n)+3f(n-1)$
   b) $f(n+1)=f(n)^2f(n-1)$
   c) $f(n+1)=3f(n)^2-4f(n-1)^2$
   d) $f(n+1)=f(n-1)/f(n)$
4. 求出$f(2)$，$f(3)$，$f(4)$和$f(5)$，若f递归地定义成$f(0)=f(1)=1$，而且对$n=1,2,\cdots$来说
   a) $f(n+1)=f(n)-f(n-1)$
   b) $f(n+1)=f(n)f(n-1)$
   c) $f(n+1)=f(n)^2+f(n-1)^3$
   d) $f(n+1)=f(n)/f(n-1)$
5. 给出序列$\{a_n\}$的递归定义，$n=1,2,3,\cdots$，若
   a) $a_n=6n$　　b) $a_n=2n+1$
   c) $a_n=10^n$　　d) $a_n=5$

6. 给出序列 $\{a_n\}$ 的递归定义，$n=1, 2, 3, \cdots$，若

   a) $a_n = 4n - 2$　　b) $a_n = 1 + (-1)^n$

   c) $a_n = n(n+1)$　　d) $a_n = n^2$

7. 设 F 是这样的函数，使得 $F(n)$ 是前 n 个正整数之和。给出 $F(n)$ 的递归定义。
8. 给出 $S_m(n)$ 的递归定义，即整数 m 与非负整数 n 之和。
9. 给出 $P_m(n)$ 的递归定义，即整数 m 与非负整数 n 之积。

在练习 10～17 里 f_n 是第 n 个斐波那契数。

10. 证明：每当 n 是正整数时，就有 $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n f_{n+1}$。
11. 证明：每当 n 是正整数时，就有 $f_1 + f_3 + \cdots + f_{2n-1} = f_{2n}$。

*12. 证明：每当 n 是正整数时，就有 $f_{n+1}f_{n-1} - f_n^2 = (-1)^n$。

*13. 证明：每当 n 是正整数时，就有 $f_0 f_1 + f_1 f_2 + \cdots + f_{2n-1} f_{2n} = f_{2n}^2$。

*14. 证明：每当 n 是正整数时，就有 $f_0 - f_1 + f_2 - \cdots - f_{2n-1} + f_{2n} = f_{2n-1} - 1$。

15. 确定欧几里德算法求出斐波那契数 f_n 和 f_{n+1} 的最大公因子所用的除法次数，其中 n 是非负整数。用数学归纳法验证你的答案。
16. 设

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

证明：每当 n 是正整数时，就有

$$\mathbf{A}^n = \begin{bmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{bmatrix}$$

17. 通过在练习 16 里等式的两边取行列式，证明练习 12 里给出的恒等式。（本题依赖于 2×2 矩阵的行列式的概念。）

*18. 给出函数 max 和 min 的递归定义，使得 $\max(a_1, a_2, \cdots, a_n)$ 和 $\min(a_1, a_2, \cdots, a_n)$ 分别是 n 个数 $a_1, a_2, \cdots, a_n$ 中的最大值和最小值。

*19. 设 $a_1, a_2, \cdots, a_n$ 和 $b_1, b_2, \cdots, b_n$ 都是实数。用你在习题 18 里给出的递归定义来证明下面的结果。

   a) $\max(-a_1, -a_2, \cdots, -a_n) = -\min(a_1, a_2, \cdots, a_n)$

   b) $\max(a_1 + b_1, a_2 + b_2, \cdots, a_n + b_n) \leqslant \max(a_1, a_2, \cdots, a_n) + \max(b_1, b_2, \cdots, b_n)$

   c) $\min(a_1 + b_1, a_2 + b_2, \cdots, a_n + b_n) \geqslant \min(a_1, a_2, \cdots, a_n) + \min(b_1, b_2, \cdots, b_n)$

20. 证明：集合 S 是正整数集合。它定义成：$1 \in S$ 而且每当 $s \in S$ 和 $t \in S$ 时就有 $s + t \in S$。
21. 给出是 5 的倍数的正整数的集合的递归定义。
22. 给出下述集合的递归定义：

   a) 正奇数集合。

   b) 3 的正整数次幂的集合。

   c) 整系数多项式的集合。

23. 给出下述集合的递归定义：

a）偶数集合。

b）与2模3同余的正整数的集合。

c）不能被5整除的正整数的集合。

24. 证明：由数字、变量、以及 {+，-，*，/，↑} 里的运算符所组成的任何合式公式，都包含同样数目的左括号和右括号。

25. 定义由集合表示集合的变量和 {—，∪，∩，-} 里的运算符所组成的合式公式。

一个字符串的倒置，是由原字符串里的符号以相反顺序组成的字符串。把字符串 w 的倒置表示成 w^R。

26. 求出下面的位串的倒置。

a）0101

b）11011

c）100010010111

27. 给出字符串的倒置的递归定义。[提示：首先定义空串的倒置。然后把长度为 $n+1$ 的字符串 w 写成 xy，其中 x 是长度为 n 的字符串，并且利用 x^R 和 y 来表示 w 的倒置。]

*28. 给出 $(w_1w_2)^R = w_2{}^R w_1{}^R$ 的递归证明。

29. 给出 w^i 的递归定义，其中 w 是字符串而 i 是非负整数。（在这里 w^i 表示字符串 w 的 i 份复制品的连接。）

*30. 给出回文位串的集合的递归定义。

31. 位串集合 A 定义成

$$\lambda \in A;$$
$$\text{若 } x \in A \text{，则 } 0x1 \in A$$

其中 λ 是空串。哪些位串属于 A？

*32. 递归地定义包含的0比1多的位串的集合。

33. 用习题29和数学归纳法证明：$l(w^i) = i \cdot l(w)$，其中 w 是位串而 i 是非负整数。

*34. 证明：每当 w 是位串而 i 是非负整数时，就有 $(w^R)^i = (w^i)^R$；即证明一个字符串的倒置的 i 次幂是这个字符串的 i 次幂的倒置。

*35. 正整数 n 的分拆是把 n 写成正整数之和的方式。例如，$7=3+2+1+1$ 是7的分拆。设 P_m 等于 m 的不同分拆的数目，其中和式里项的顺序无关紧要，并设 $P_{m,n}$ 是把用不超过 n 的正整数来表示 m 的不同方式的数目。

a）证明：$P_{m,m} = P_m$。

b）证明：下面的 $P_{m,n}$ 的递归定义是正确的：

$$P_{m,n} = \begin{cases} 1, & \text{若 } m=1 \\ 1, & \text{若 } n=1 \\ P_{m,n}, & \text{若 } m<n \\ 1+P_{m,m-1}, & \text{若 } m=n>1 \\ P_{m,n-1}+P_{m-n,n}, & \text{若 } m>n>1 \end{cases}$$

c）用这个递归定义求出5和6的分拆数。

考虑阿克曼函数的一个变种的下述递归定义。这个函数是根据德国数学家威尔海姆·阿克曼的名字来命名的，他是大数学家大卫·希尔伯特的学生。在递归函数论中以及在涉及集合的并的某些算法的复杂性研究中，阿克曼函数都起到重要的作用。（这个函数有多种不同的变种，都称为阿克曼函数，并且都有类似的性质，尽管它们的值不一定是相等的。）

$$A(m,n)=\begin{cases}2n, & \text{若 } m=0\\ 0, & \text{若 } m\geqslant 1 \text{ 且 } n=0\\ 2, & \text{若 } m\geqslant 1 \text{ 且 } n=1\\ A(m-1,A(m,n-1)), & \text{若 } m\geqslant 1 \text{ 且 } n\geqslant 2\end{cases}$$

练习 36～43 涉及这种形式的阿克曼函数。

36. 求出阿克曼函数的下列值。
  a) A (1, 0) b) A (0, 1) c) A (1, 1) d) A (2, 2)
37. 证明：每当 $m\geqslant 1$ 时，就有 $A(m,2)=4$。
38. 证明：每当 $n\geqslant 1$ 时，就有 $A(1,n)=2^n$。
39. 求出阿克曼函数的下列值。
  a) $A(2,3)$ *b) $A(3,3)$
*40. 求出 $A(3,4)$。
**41. 证明：每当 m 和 n 都是非负整数时，就有 $A(m,n+1)\geqslant A(m,n)$。
*42. 证明：每当 m 和 n 都是非负整数时，就有 $A(m+1,n)\geqslant A(m,n)$。
43. 证明：每当 i 和 j 都是非负整数时，就有 $A(i,j)\geqslant j$。
☞44．用数学归纳法证明：通过规定 $F(0)$ 和从 $F(n)$ 获得 $F(n+1)$ 的规则，所定义的函数 F 是严格定义的。
☞45．用数学归纳法第二原理证明：通过规定 F (0)，以及对 $k=0, 1, 2, \cdots, n$ 来说从 F (k) 获得 F ($n+1$) 的规则，所定义的函数是严格定义的。
46. 证明：下述每一个所谓的对正整数集合上的函数的递归定义都不能产生严格定义的函数。
  a) 对 $n\geqslant 1$ 来说 $F(n)=1+F(\lfloor n/2\rfloor)$，而且 $F(1)=1$。
  b) 对 $n\geqslant 2$ 来说 $F(n)=1+F(n-3)$，而且 $F(1)=2$ 和 $F(2)=3$。
  c) 对 $n\geqslant 2$ 来说 $F(n)=1+F(n/2)$，而且 $F(1)=1$ 和 $F(2)=2$。
  d) 若 n 是偶数而且 $n\geqslant 2$，则 $F(n)=1+F(n/2)$，若 n 是奇数，则 $F(n)=1-F(n-1)$，以及 $F(1)=1$。
  e) 若 n 是偶数而且 $n\geqslant 2$，则 $F(n)=1+F(n/2)$，若 n 是奇数而且 $n\geqslant 3$，则 $F(n)=F(3n-1)$，以及 $F(1)=1$。
47. 证明：下述每一个所谓的对正整数集合上的函数的递归定义都不能产生严格定义的函数。
  a) 对 $n\geqslant 1$ 来说 $F(n)=1+F(\lfloor(n+1)/2\rfloor)$，而且 $F(1)=1$。
  b) 对 $n\geqslant 2$ 来说 $F(n)=1+F(n-2)$，而且 $F(1)=0$
  c) 对 $n\geqslant 3$ 来说 $F(n)=1+F(n/3)$，而且 $F(1)=1, F(2)=2$，以及 $F(3)=3$。
  d) 若 n 是偶数而且 $n\geqslant 2$，则 $F(n)=1+F(n/2)$，若 n 是奇数，则 $F(n)=1+F(n-2)$，

而且$F(1)=1$。

e) 若 $n\geqslant 2$ 则$F(n)=1+F(F(n-1))$，而且$F(1)=2$。

练习 48～50 处理对数函数的迭代。像通常一样，设 $\log n$ 表示以 2 为底 n 的对数。函数 $\log^{(k)} n$ 递归地定义成

$$\log^{(k)} n=\begin{cases} n, & \text{若 } k=0 \\ \log\left(\log^{(k-1)} n\right), & \text{若 } \log^{(k-1)} n \text{ 有定义且为正数} \\ \text{无定义}, & \text{其他情况} \end{cases}$$

迭代对数是函数$\log^* n$，它在 n 处的值是使得 $\log^{(k)} n\leqslant 1$ 的最小的非负整数 k。

48. 求出下述的每一个值：
    a) $\log^{(2)}16$　b) $\log^{(3)}256$　c) $\log^{(3)}2^{65536}$　d) $\log^{(4)}2^{65536}$
49. 对下述的每一个 $\log^* n$ 的值，求出 n 的值：
    a) 2　b) 4　c) 8　d) 16　e) 256　f) 65536　g) 2^{2048}
50. 求出使得 $\log^* n=5$ 的最大整数 n。确定这个数的十进制位数。

练习 51～53 处理迭代函数的值。假定$f(n)$是从实数集合、正实数集合或某些其他的实数集合到实数集合的函数，使得$f(n)$是单调增的［即当 $n<m$ 时有 $f(n)<f(m)$，并且对 f 的定义域里的所有 n 来说$f(n)<n$。］函数$f^{(k)}(n)$递归地定义成

$$f^{(k)}(n)=\begin{cases} n, & \text{若 } k=0 \\ f(f^{(k-1)}(n)), & \text{若 } k>0 \end{cases}$$

另外，设 c 是正实数。迭代函数f^*_c是为了把f的自变量缩小到小于或等于c所需要的f的迭代次数，所以$f^*_c(n)$是使得 $f^{(k)}(n)\leqslant c$ 的最小的非负整数k。

51. 设 $f(n)=n-a$，其中 a 是正整数。求出$f^{(k)}(n)$的公式。当 n 是正整数时，$f_0^*(n)$的值是什么？
52. 设$f(n)=n/2$。求出 $f^{(k)}(n)$的公式。当 n 是正整数时，$f_1^*(n)$ 的值是什么？
53. 设$f(n)=\sqrt{n}$。求出 $f^{(k)}(n)$的公式。当 n 是正整数时，$f_2^*(n)$的值是什么？

练习 54～61 处理某些不寻常的序列，这些序列非正式地称为*自生成序列*，它们是用简单的递归关系或规则产生的。具体地说，练习 54～57 处理序列 $\{a(n)\}$，它定义成：对$n\geqslant 1$来说$a(n)=n-a(a(n-1))$，而且 $a(0)=0$。（这个序列，以及在练习 58 和练习 59 里的序列，都是在道格拉斯·霍夫斯塔德的奇妙的书《歌德尔、埃舍尔、巴赫》［Ho99］中定义的）。

54. 求出在本题前面的说明中定义的序列 $\{a(n)\}$ 的前 10 项。

*55. 证明：这个序列是严格定义的。即证明对所有非负整数 n 来说 $a(n)$ 是唯一地定义的。

**56. 证明：$a(n)=\lfloor(n+1)\mu\rfloor$，其中$\mu=(-1+\sqrt{5})/2$。［提示：首先证明对所有 $n>0$ 来说$(\mu n-\lfloor\mu n\rfloor)+(\mu^2\lfloor\mu^2 n\rfloor)=1$。然后证明对满足 $0\leqslant\alpha<1$ 和 $\alpha\neq 1-\mu$ 的所有实数α来说$\lfloor(1+\mu)(1-\alpha)\rfloor+\lfloor\alpha+\mu\rfloor=1$，分别考虑 $0\leqslant\alpha<1-\mu$ 和 $1-\mu<\alpha<1$ 的情形。

*57. 利用练习 56 的公式证明：若 $\mu n - \lfloor \mu n \rfloor < 1 - \mu$ 则 $a(n) = a(n-1)$，否则 $a(n) = a(n-1)+1$。

58. 求出下面每个自生成序列的前 10 项：

a) 对 $n \geq 1$ 来说 $a(n) = n - a(a(a(n-1)))$，而且 $a(0) = 0$。

b) 对 $n \geq 1$ 来说 $a(n) = n - a(a(a(a(n-1))))$，而且 $a(0) = 0$。

c) 对 $n \geq 3$ 来说 $a(n) = a(n - a(n-1)) + a(n - a(n-2))$，而且 $a(1) = 1$ 和 $a(2) = 1$。

59. 求出序列 $m(n)$ 和 $f(n)$ 的前 10 项，它们是用下面一对交织的递归关系来定义的：对 $n \geq 1$ 来说 $m(n) = n - f(m(n-1))$，$f(n) = n - m(f(n-1))$，而且 $f(0) = 1$ 和 $m(0) = 0$。

哥伦布的自生成序列是具有下述性质的唯一非减的正整数序列 $a_1, a_2, a_3, \cdots$：对每个正整数 k 来说这个序列恰好包含 k 的 a_k 次出现。

60. 求出哥伦布的自生成序列的前 20 项。

*61. 证明：若 $f(n)$ 是使得 $a_m = n$ 的最大整数 m，其中 a_m 是哥伦布的自生成序列的第 m 项，则 $f(n) = \sum_{k=1}^{n} a_k$，而且 $f(f(n)) = \sum_{k=1}^{n} k a_k$。

G.H.哈代[○]引入的对数指数函数集合 $\mathcal{L}$ 是使得下面性质成立的最小的函数集合：

○ 葛弗雷·哈罗得·哈代（Godfrey Harold Hardy，1877—1947） 哈代出生在英格兰的色雷的克兰雷，是伊萨克·哈代与索菲亚·哈尔·哈代的两个孩子中的长子。他的父亲是克兰雷中学的地理学与绘图教师，还教过唱歌课和踢过足球。他的母亲教钢琴课而且帮助经营为年轻学生提供膳宿的宿舍。哈代的父母亲对子女的教育全力以赴。当哈代在两岁开始写出表示上百万的数字时，就显示了他的数字能力。他有一个私人数学教师而不是上克兰雷中学的正规课程。当他十三岁时，他转到温彻斯特学校，被授予奖学金，这是一所私立高中。他学习成绩优秀，显示出对数学的强烈兴趣。他在 1896 年获得奖学金进入剑桥的三一学院，在学期间几次获奖，于 1899 年毕业。

哈代从 1906 年到 1919 年在剑桥大学三一学院担任数学讲师职务，然后被任命为牛津的几何学的苏里文教授。他在剑桥变得很不愉快，原因是在反战活动上与三一学院的著名哲学家和数学家伯特兰·罗素不和，以及不喜欢繁重的行政工作。在 1931 年他返回剑桥，担任纯数学的萨德雷里安教授，一直到 1942 年退休为止。他是一个纯数学家并对数学持有卓尔不群观点，希望他的研究永远不被应用。具有讽刺意味的是，他最出名的也许是作为哈代-温伯格法则的发明人之一，这个法则预言遗传模式。他在这个领域的工作是作为给《科学》杂志的一封信出现的，在信中他用简单的代数想法来指出一篇遗传学论文的错误。哈代主要在数论和函数论方面做工作，研究黎曼 zeta 函数、傅立叶级数、素数分布这样的课题。他对许多重要问题作出了重要的贡献，比如关于把正整数表示成 k 次幂之和的华林问题，以及把奇数表示成三个素数之和的问题。哈代还因为与剑桥的同事约翰·E·利特伍德和著名的印度数学奇才斯里尼瓦萨·拉马努金[○]的合作而被人们记忆。他与利特伍德合写了超过一百篇论文。他们的合作引出这样的笑话：在当时只有三个重要的英国数学家：哈代、利特伍德和哈代-利特伍德，甚至有人认为哈代发明了虚构的人物利特伍德，因为在剑桥以外很少看见利特伍德。哈代具有这样的智慧，他从拉马努金寄给他的非常规的但是极有创造性的结果中，识别出拉马努金的天才，而其他数学家却没能看出拉马努金的天才。哈代把拉马努金带到剑桥，合作写出重要的联名文章，证明了关于整数的分拆数的新结果。哈代对数学教育感兴趣，他的书《纯数学教程》在二十世纪的前五十年里深深地影响了数学的本科教学。哈代还写过《一个数学家的辩解》，在这本书里他回答了这样的问题：一个人是否值得把一生都献给数学研究。这本书给出了哈代关于数学是什么和数学做什么的观点。

哈代对体育有强烈兴趣。他是狂热的板球迷，对分数盯得很紧。他的一个独特的习惯是，他不喜欢拍照（只有五张照片是为人所知的），也不喜欢镜子，他一进入旅馆房间就立即用毛巾把它们盖上。

- 函数$f(n)=\alpha$属于$\mathscr{L}$，其中α是一个正实数；
- 函数$f(n)=n$属于$\mathscr{L}$；
- 若函数$f(n)$和$g(n)$都属于$\mathscr{L}$,则$f(n)-g(n)$属于$\mathscr{L}$；
- 若函数$f(n)$属于$\mathscr{L}$,则$e^{f(n)}$属于$\mathscr{L}$；
- 若$f(n)$属于$\mathscr{L}$，而且存在整数N使得对$n\geqslant N$来说$f(n)>0$（这意味着f是最终成为的），则$\ln f(n)$属于$\mathscr{L}$，其中$\ln x$与通常一样表示x的对数。

哈代证明，对每个不恒等于0的对数指数函数来说，$f(n)$或者是最终成为正的，或者是最终成为负的。他证明若$f(n)$和$g(n)$都属于$\mathscr{L}$,则或者$f(n)$是$o(g(n))$,或者$g(n)$是$o(f(n))$,或者$f(n)$和$g(n)$有同样的阶。

62. 证明：若$f(n)$和$g(n)$都属于$\mathscr{L}$，则$f(n)+g(n)$属于$\mathscr{L}$。

*63. 证明：若$f(n)$和$g(n)$都属于$\mathscr{L}$而且都是最终成为正的,则$f(n)g(n)$和$f(n)/g(n)$都属于$\mathscr{L}$，利用每个对数指数函数都是或者最终成为正、最终成为负或者恒等于0的事实来证明，前面的结果蕴涵着任何两个不恒等于0的属于$\mathscr{L}$的函数之积与商都属于$\mathscr{L}$。

64. 利用练习62和练习63证明：每个带实系数的多项式$f(n)=a_mn^m+a_{m-1}a^{m-1}+\cdots+a_0$属于$\mathscr{L}$。

㊀ 斯里尼瓦萨·拉马努金（Srinivasa Ramanujan，1887—1920） 著名的数学奇才拉马努金诞生和成长在靠近马德拉斯城的南印度。他的父亲是布店的职员。他的母亲在地方上的寺庙里唱歌来增加家庭收入。拉马努金在地方上的英语学校学习，显示了他对数学的天才和兴趣。在十三岁时他就掌握了大学生使用的教科书。当他十五岁时，一位大学生借给他一本《纯数学提要》。拉马努金决心做完这本书里只有叙述而没有证明和解释的六千多个结果，他写在纸片上，以后收集起来成了笔记本。他在1904年从高中毕业，赢得马德拉斯大学的奖学金。他注册了美术专业的课程，但是他只顾数学而忽略了他的专业，结果丢了奖学金。从1904年到1907年他四次在大学里考试不及格，只在数学上成绩好。在这段时间里他在笔记本里写下了创造性的结果，有时重新发现了已经发表过的结果，而其他时候则做出了新的发现。

由于没有大学学位，拉马努金难以找到好的工作。为了活命，他不得不依赖朋友们的施舍。他教过学生数学，但是他非同寻常的思考方式和不遵守教学大纲却引来麻烦。他在1909年在包办婚姻中与比他小九岁的年轻女子结婚。为了养活自己和妻子，他搬到马德拉斯并且找了个工作。他向可能雇佣他的人出示他的数学结果的笔记本，但是笔记本却惹恼了他们。不过，总统学院的一位教授看出他的天才并且支持过他一段时间，在1912年他找到做会计员的工作，挣微薄的薪水。

拉马努金在这段时间里继续他的数学工作，并且在1910年在一份印度期刊上发表了他的第一篇论文。他认识到他的工作超过了印度数学家的理解范围，决定给最重要的英国数学家写信。第一个收到信的数学家拒绝了他寻求帮助的要求，但是在1913年元月他写信给哈代，哈代曾想拒绝拉马努金，但是信中的数学命题却让哈代感到迷惑，尽管没有证明。他决定在同事与合作者利特伍德的帮助下仔细地检查它们。在仔细研究之后，他们断定拉马努金可能是个天才，因为他的那些命题"只能是第一流的数学家写出来的；它们一定是对的，因为假如它们不对，那么就没有人具有发明它们的想象力。"

哈代为拉马努金安排了奖学金，在1914年把他带到英格兰。哈代亲自教他数学分析，他们合作了五年，证明了关于整数的分拆数的重要定理。在这段时间里，拉马努金对数论作出了重要贡献，还在连分数、无穷级数以及椭圆曲线方面做了工作。拉马努金对特定类型的函数和级数具有惊人的洞察力，但是他关于素数的所谓的定理却常常是错的，暴露了他对什么是正确证明的模糊观点。他是被任命为皇家协会成员的最年轻的成员之一。不幸的是，在1917年拉马努金病得很重，当时，他被认为不适应英国的气候并且感染了肺结核。现在则认为他患的是维生素缺乏症，起因是拉马努金严格的素食主义和战时英国的食物匮乏。他在1919年回到印度，即使当他离不开床时，也继续研究数学。他是教徒并且认为他的数学才能来自家庭的守护神——拉马吉里。他认为数学与宗教是联系的。他说："一个等式对我来说毫无意义，除非它表达了神的思想。"1920年四月他短暂的生命走到了终点，当时他32岁。拉马努金留下了写有未发表结果的几个笔记本。这些笔记本里的结果说明了拉马努金的洞察力，但都是快速的勾勒。若干数学家花了许多年的研究来解释和证实这些笔记本里的结果。

65．证明：若 $f(n)$ 属于 $\mathcal{L}$，并且是最终成为正的，则 $\sqrt{f(n)}$ 属于 $\mathcal{L}$。

66．证明：函数 $e^{\sqrt{n}\sqrt{\ln n \ln\ln n}}$ 属于 $\mathcal{L}$。

3.4　递归算法

3.4.1　引言

有时可以把带有具体的一组输入的问题的解归约到带更小的一组输入的相同问题的解。例如，求两个正整数 a 和 b 的最大公因子的问题，其中 $b>a$，就可以归约到求一对更小的整数（即 b mod a 和 a）的最大公因子的问题，因为 $\gcd(b \bmod a, a)=\gcd(a,b)$。当可以实现这样的归约时，就可以用一系列归约来求出原问题的解，直到把问题归约到解是已知的某个初始情形为止。例如，对求最大公因子来说，归约持续到两个数中较小的一个为零，因为当 $a>0$ 时，$\gcd(a,0)=a$。

将要看出，连续地把问题归约到带更小的输入的相同问题，这样的算法可用来解决广泛的问题。

定义 1　若一个算法通过把问题归约到带更小的输入的相同问题的实例，来解决原来的问题，则这个算法称为*递归的*。

在下面的例子里将描述几个不同的递归算法。第一个例子说明如何构造递归算法从函数的递归定义来求函数的值。

例 1　给出计算 a^n 的递归算法，其中 a 是非零实数而 n 是非负整数。

解　可以让递归算法是基于 a^n 的递归定义。这个定义说对 $n>0$ 来说有 $a^{n+1}=a\cdot a^n$，而初始条件是 $a^0=1$。为了求出 a^n，就连续地用这个递归定义来缩小指数，直到指数是零为止。在算法 1 里给出这个过程。■

算法 1　计算 a^n 的递归算法

```
procedure power(a:非零实数,n:非负整数)
if n = 0 then power(a,n):= 1
else power(a,n):= a * power(a,n - 1)
```

下一步给出求最大公因子的递归算法。

例 2　给出求满足 $a<b$ 的两个非负整数 a 和 b 的最大公因子的递归算法。

解　可以让递归算法是基于归约 $\gcd(a,b)=\gcd(b \bmod a, a)$ 和条件当 $b>0$ 时 $\gcd(0,b)=b$。这样就产生了算法 2 里的过程。■

算法 2　计算 $\gcd(a,b)$ 的递归算法

```
procedure gcd(a,b:非负整数且 a<b)
if a = 0 then gcd(a,b):= b
else gcd(a,b):= gcd(b mod a,a)
```

现在将要给出搜索算法的递归形式。

例 3 把线性搜索算法表达成递归过程。

解 为了在搜索序列 a_1，a_2，…，a_n 里搜索 x，在算法的第 i 步比较 x 与 a_i。若 x 等于 a_i，则 i 是 x 的位置。否则，对 x 的搜索就归约到在少了一个元素的序列（即序列 a_{i+1}，…，a_n）里的搜索。现在给出递归过程。

设 $search(i,j,x)$ 是在序列 a_i，a_{i+1}，…，a_j 里搜索 x 的过程。过程的输入包括三元组 $(1, n, x)$。若剩余序列的第一项是 x，或者若序列只有一项并且它不是 x，则过程在这一步终止。若 x 不是第一项而且存在其他的项，则执行同样的过程，但是搜索序列减少一项，它是通过删除搜索序列的第一项而获得的。 ■

算法 3 递归顺序搜索算法

```
procedure search(i, j, x)
if a_i = x then
    location := i
else if i = j then
    location := 0
else
    search(i + 1, j, x)
```

例 4 构造二叉搜索算法的递归形式。

解 假定想在序列 a_1，a_2，…，a_n 里求出 x 的位置。为了执行二叉搜索，首先比较 x 与中间项 $a_{\lfloor (n+1)/2 \rfloor}$。若 x 等于这一项，则算法将终止。否则，把搜索归约到更小的搜索序列，即若 x 小于原序列的中间项，则归约到序列的前一半，否则归约到后一半。已经把搜索问题的解归约到带长度近似为一半的序列的相同问题的解。把二叉搜索算法的这种递归形式表达成算法 4。 ■

算法 4 递归二叉搜索算法

```
procedure binary search(x, i, j)
m := ⌊(i + j)/2⌋
if x = a_m then
    location := m
else if (x < a_m 且 i < m) then
    binary search(x, i, m - 1)
else if (x > a_m 且 j > m) then
    binary search(x, m + 1, j)
else location := 0
```

3.4.2 递归与迭代

递归定义把在正整数处的函数值表达成在更小的整数处的函数值。这意味着可以设计递

归算法来求出递归地定义的函数在正整数处的值。

例 5 下面的递归过程当输入是正整数 n 时给出 $n!$ 的值。■

算法 5 阶乘的递归过程

```
procedure factorial(n:正整数)
if n = 1 then
    factorial(n):= 1
else
    factorial(n):= n * factorial(n - 1)
```

存在另外一种方式，从阶乘函数的递归定义求它在整数处的值。代替连续地把计算归约到在更小的整数处来求函数的值，可以从在 1 处的函数值开始，连续地应用递归定义来求出在更大的整数处的函数值。这样的过程称为迭代。换句话说，为了用迭代过程求出 $n!$，从 1（即在 1 处的阶乘函数值）开始，连续地乘以每个小于或等于 n 的正整数。这个过程显示在算法 6 里。

算法 6 阶乘的迭代过程

```
procedure iterative factorial(n:正整数)
x := 1
for i := 1 to n
    x := i * x
{x is n!}
```

在执行了这段代码之后，变量 x 的值是 $n!$。例如，执行这个循环六遍之后给出 $6! = 1\cdot2\cdot3\cdot4\cdot5\cdot6 = 720$。

对递归地定义的序列求值的迭代方法，比起使用递归的过程来，常常要求较少量的计算（除非使用专门的递归机器）。用求第 n 个斐波那契数的迭代和递归过程来说明这一点。首先给出递归过程。

算法 7 斐波那契数的递归算法

```
procedure fibonacci(n:非负整数)
if n = 0 then fibonacci(0):= 0
else if n = 1 then fibonacci(1):= 1
else fibonacci(n):= fibonacci(n - 1) + fibonacci(n - 2)
```

当使用递归算法求 f_n 时，首先把 f_n 表示成 $f_{n-1}+f_{n-2}$。然后把这两个斐波那契数都换成两个前面的斐波那契数之和。当 f_0 或 f_1 出现时，就直接换成它的值。

注意，在递归的每个阶段，直到获得 f_1 或 f_0 为止，需要求值的斐波那契数的个数都一

直翻倍。例如，当使用这个递归算法求出 f_4 时，就必须完成图 3-10 里的树形图所说明的全部计算。这个树包括用 f_4 标记的根，以及从根到用两个斐波那契数 f_3 和 f_2 标记的顶点的分支，它们出现在 f_4 的计算的归约里。每个后续的归约都产生树里的两个分支。当遇到 f_0 和 f_1 时，这种分支结束。读者可以验证一下，这个算法需要 $f_{n+1}-1$ 次加法来求出 f_n。

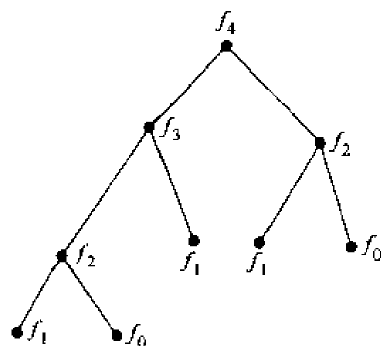


图 3-10 递归地对 f_4 求值

现在考虑用下面的迭代过程来求出 f_n 所需要的计算量。

算法 8 计算斐波那契数的迭代算法

```
procedure iterative fibonacci(n:非负整数)
if n = 0 then y := 0
else
begin
    x := 0
    y := 1
    for i := 1 to n - 1
    begin
        z := x + y
        x := y
        y := z
    end
end
{y 是第 n 个斐波那契数}
```

这个过程把 x 初始化成 $f_0=0$，把 y 初始化成 $f_1=1$。当经过循环时，把 x 和 y 之和赋给辅助变量 z。然后把 x 赋成 y 的值，而把 y 赋成辅助变量 z 的值。因此，在经过第一次循环之后得出 x 等于 f_1 而 y 等于 $f_0+f_1=f_2$。另外，在经过 $n-1$ 次循环之后 x 等于 f_{n-1}，而且 y 等于 f_n（读者应当验证这个命题）。当 $n>1$ 时，用这个迭代方法求出 f_n 仅仅使用了 $n-1$ 次加法。因此，这个算法比递归算法需要少得多的计算。

已经说明当求递归定义的函数的值时，递归算法可能比迭代算法需要多得多的计算量。有时宁愿使用递归算法，即使它比迭代过程更低效。特别是，当递归方法容易实现而迭代方法不容易实现时，这种说法就是对的。（另外，或许可以用专门设计用来处理递归的机器，它们抵消了使用迭代的好处。）

练习

1. 给出每当 n 是正整数而 x 是整数时，计算 nx 的递归算法。
2. 给出求前 n 个正整数之和的递归算法。

3. 给出求前 n 个正奇数之和的递归算法。
4. 给出求一个有限个整数集的最大值的递归算法。
5. 给出求一个有限整数集的最小值的递归算法。
6. 设计一个递归算法，每当 n，x，和 m 都是正整数时，它求出 $x^n \bmod m$。
7. 给出每当 n 和 m 都是正整数时，求 $n! \bmod m$ 的递归算法。
8. 给出求出整数列表中的众数的递归算法。(众数是列表中出现得至少像其他每个元素一样频繁的元素。)
9. 设计一个递归算法，它计算满足 $a<b$ 的两个非负整数 a 和 b 的最大公因子，假如 $\gcd(a, b)=\gcd(a, b-a)$。
10. 设计求 a^{2^n} 的递归算法，其中 a 是实数而 n 是正整数。[提示：利用等式 $a^{2^{n+1}}=(a^{2^n})^2$。]
11. 对于求 a^{2^n} 的值，练习 10 里的算法所用的乘法次数与算法 1 所用的乘法次数相比较的结果如何？

*12. 用练习 10 里的算法，设计当 n 是非负整数时求 a^n 的值的算法。[提示：利用 n 的二进制展开式。]

*13. 对于求 a^n 的值，练习 12 里的算法所用的乘法次数与算法 1 所用的乘法次数相比较的结果如何？

14. 为了求出斐波那契数 f_7，在算法 7 和算法 8 里给出的递归算法和迭代算法，各自分别使用多少次加法？
15. 设计求一个序列的第 n 项的递归算法，该序列定义成：$a_0=1$，$a_1=2$，而且对 $n=2$，3，4，…来说有 $a_n=a_{n-1}\cdot a_{n-2}$。
16. 设计求练习 15 里定义的序列的第 n 项的迭代算法。
17. 求练习 15 里的序列的递归算法与迭代算法，哪个算法更有效？
18. 设计求一个序列的第 n 项的递归算法，该序列定义成 $a_0=1$，$a_1=2$，$a_2=3$，而且对 $n=3$，4，5，…来说有 $a_n=a_{n-1}+a_{n-2}+a_{n-3}$。
19. 设计求练习 18 里定义的序列的第 n 项的迭代算法。
20. 求练习 18 里的序列的递归算法与迭代算法，哪个算法更有效？
21. 给出求一个序列的第 n 项的递归算法和迭代算法，该序列定义成 $a_0=1$，$a_1=3$，$a_2=5$，而且 $a_n=a_{n-1}\cdot a_{n-2}^2\cdot a_{n-3}^3$。哪个算法更有效？
22. 根据 3.3 节练习 35 里给出的递归定义，给出求正整数的划分数的递归算法。
23. 给出求字符串的倒置的递归算法。(见 3.3 节练习 26 前面的说明里对位串的倒置的定义。)
24. 给出当 w 是位串时，求字符串 w^i（即 w 的 i 个复制品的连接）的递归算法。
25. 给出计算阿克曼函数值的递归算法。[提示：见 3.3 节里练习 36 前面的说明。]

3.5　程序正确性

3.5.1　引言

假定设计了解决一个问题的算法，而且编写了实现它的程序。如何才能保证这个程序总是产生正确的答案？在消除了所有的错误使得语法是正确的之后，可以用简单的输入来测试

这个程序。若对任何简单输入来说产生了不正确的结果，则它是不正确的。但是即使对所有的简单输入来说这个程序都给出了正确的答案，它也不一定总是产生正确的答案（除非已经测试了所有可能的输入)。我们需要一个说明这个程序总是给出正确答案的证明。

程序验证（即程序正确性的证明）使用在本章里描述的推理规则和证明技术，包括数学归纳法。因为不正确的程序可能导致灾难性的后果，所以已经构造了大量的方法来对程序进行验证。已经作出让程序验证自动化以便可以用计算机来完成它的努力。不过，朝向这个目标才仅仅取得了有限的进展。事实上，一些数学家和理论计算机科学家争论说让复杂程序的正确性证明机械化将永远是不现实的。

本节将介绍用来证明程序为正确的一些概念和方法。不过，在本书里将不发展程序验证的完整方法。本节打算成为对程序验证领域的粗略介绍，它把逻辑规则、证明技术以及算法的概念联系到了一起。

3.5.2 程序验证

若对每个可能的输入来说一个程序都产生正确的输出，则说这个程序是*正确的*。一个程序为正确的证明包括两个部分。第一部分证明：若程序终止，则获得正确的答案。证明的这一部分确立了程序的*部分正确性*。证明的第二部分表明程序总是终止。

为了规定程序产生正确的输出是什么意思，使用两个命题。第一个是*初始断言*，它给出输入值必须具有的性质。第二个是*终结断言*，它给出假如程序做了要求它做的事情，则程序的输出应当具有的性质。当验证一个程序时，必须提供适当的初始断言和终结断言。

定义 1 若每当对一个程序或程序段 S 的输入值来说初始断言 p 为真时，就有对 S 的输出值来说终结断言 q 为真，则说 S 是相对于 p 和 q 部分正确的。记号 $p\{S\}q$ 说明程序或程序段 S 是相对于初始断言 p 和终结断言 q 部分正确的。注释：记号 $p\{S\}q$ 称为霍尔三元组，因为托尼·霍尔[⊖]引入了部分正确性的概念。

注意，部分正确性的概念与程序是否终止是无关的；它仅仅关注若程序终止，则程序是否做了期待它做的事情。

一个简单的例子可以说明初始断言和终结断言的概念。

例 1 证明程序段

$y := 2$
$z := x + y$

是相对于初始断言 $p: x = 1$ 和终结断言 $q: z = 3$ 是正确的。

解 假定 p 为真，所以在程序开始时 $x = 1$。则把 y 赋值成 2，而把 z 赋值成 x 和 y 值之和，即 3。因此，S 是相对于初始断言 p 和终结断言 q 是正确的。因此，$p\{S\}q$ 为真。 ■

⊖ 安托尼·霍尔（C. Anthony R. Hoare，1934 年生） 霍尔目前是英格兰牛津大学计算机科学教授，以及英国皇家学会成员。霍尔对编程语言的理论和编程方法学作出了许多重要贡献。基于如何证明程序相对于它们的规格说明来说为正确的来定义编程语言，他是第一个这样做的人。霍尔也是快速排序的发明者，这是被最普遍地使用和研究的排序算法之一（见 8.4 节的练习)。霍尔是计算机科学的技术方面和社会方面的著名作家。

3.5.3 推理规则

一条有用的推理规则通过把一个程序分成一系列子程序，然后证明每个子程序为正确的来证明这个程序为正确的。

假定把程序 S 分成子程序 S_1 和 S_2。用 $S=S_1;\ S_2$ 来表示 S 是由 S_1 后接 S_2 来组成的。假定已经证明了 S_1 相对于初始断言 p 和终结断言 q 的正确性，以及 S_2 相对于初始断言 q 和终结断言 r 的正确性。由此得出，若 p 为真而且 S_1 执行且终止，则 q 为真；若 q 为真而且 S_2 执行且终止，则 r 为真。因此，若 p 为真而且 $S=S'_1;\ S_2$ 执行且终止，则 r 为真。这条推理规则称为合成规则，它可以叙述成

$$\begin{array}{l} p\{S_1\}q \\ q\{S_2\}r \\ \hline \therefore p\{S_1;S_2\}r \end{array}$$

在本节后面将使用这条推理规则。

下一步，将给出含有条件语句和循环的程序段的推理规则。因为可以把程序分成程序段，以便进行正确性证明，所以这样就能够验证许多不同的程序。

3.5.4 条件语句

首先将要给出条件语句的推理规则。假定一个程序段形如

```
If condition then
    S
```

其中 S 是一个语句块。若 *condition* 为真，则 S 执行，而当 *condition* 为假时，则 S 不执行。为了验证这个程序段是相对于初始断言 p 和终结断言 q 来说为正确的，必须做两件事情。首先，必须证明：当 p 为真而且 *condition* 也为真时，在 S 终止之后 q 为真。其次，必须证明当 p 为真而且 *condition* 为假时，q 为真（因为在这种情形里 S 不执行。）

这导致下面的推理规则：

$$\begin{array}{l} (p \wedge condition)\{S\}q \\ (p \wedge \neg\ condition) \rightarrow q \\ \hline \therefore p\{\textbf{if}\ condition\ \textbf{then}\ S\}q \end{array}$$

下面的例子说明如何使用这条推理规则。

例 2 验证程序段

```
if x > y then
    y := x
```

是相对于初始断言 **T** 和终结断言 $y \geq x$ 为正确的。

解 当初始断言为真而 $x > y$ 时，则完成赋值语句 $y := x$。因此，在这种情形里，断言 $y \geq x$ 的终结断言为真。另外，当初始断言为真而 $x > y$ 为假，因而 $x \leq y$ 时，终结断言再次

为真。因此，使用这种类型的程序段的推理规则，这个程序是相对于给定的初始断言和终结断言为正确的。■

同理，考虑含有形如

```
if condition then
    S1
else
    S2
```

的语句的程序。若 *condition* 为真，则执行 S_1；若 *condition* 为假，则执行 S_2。为了验证这个程序段是相对于初始断言 p 和终结断言 q 为正确的，必须做两件事情。首先，必须证明：当 p 为真而且 *condition* 为真时，在 S_1 终止之后 q 为真。其次，必须证明：当 p 为真而且 *condition* 为假时，在 S_2 终止之后 q 为真。这导致下面的推理规则：

$$\begin{array}{l} (p \wedge condition)\{S_1\}q \\ (p \wedge \neg\ condition)\{S_1\}q \\ \hline \therefore p\{\textbf{if}\ condition\ \textbf{then}\ S_1\ \textbf{else}\ S_2\}q \end{array}$$

下面的例子说明如何使用这条推理规则。

例 3　验证程序段

```
if x<0 then
    abs := -x
else
    abs := x
```

是相对于初始断言 T 和终结断言 $abs = |x|$ 为正确的。

解　必须证明两件事情。首先必须证明：若初始断言为真而 $x<0$，则 $abs = |x|$。这是正确的，因为当 $x<0$ 时赋值语句 $abs := -x$ 让 $abs = -x$，根据定义当 $x<0$ 时它是 $|x|$。其次必须证明：若初始断言为真而 $x<0$ 为假（所以 $x \geq 0$）则 $abs = |x|$。这是正确的，因为在这种情形里，程序使用赋值语句 $abs := x$，而根据定义，当 $x \geq 0$ 时 x 是 $|x|$，所以 $abs := x$。因此，利用对于这种类型的程序段的推理规则，这个程序是相对于给定的初始断言和终结断言为正确的。■

3.5.5　循环不变量



下一步将要描述 **while** 循环的正确性证明。为了建立一个

```
while condition
    S
```

这种类型的程序段的推理规则，要注意，S 反复执行直到 *condition* 变假为止。必须选择一个每次执行 S 时都保持为真的断言。这样的断言称为循环不变量。换句话说，若 $(p \wedge$

condition){S}p 为真，则 p 是循环不变量。

假定 p 是循环不变量。若在执行这个程序段之前 p 为真，则在程序终止后 p 和 $\neg$ *condition* 都为真，假如程序真的终止。这个推理规则是

$$\frac{(p \wedge condition)\ \{S\}\ p}{\therefore p\ \{\mathbf{while}\ condition\ S\}\ (\neg\ condition \wedge p)}$$

例 4　需要一个循环不变量来验证当 n 是正整数时，程序段

```
i := 1
factorial := 1
while i < n
begin
    i := i + 1
    factorial := factorial * i
end
```

终止而且满足 $factorial = n!$。设 p 是命题："$factorial = i!$ 并且 $i \leq n$"。我们将用数学归纳法来证明 p 是循环不变量。首先，注意在进入循环之前 p 为真，因为此时此刻 $fractorial = 1 = 1!$ 而且 $1 \leq n$。现在假定在循环执行一次之后 p 为真而且 $i < n$。假定 **while** 循环再次执行。首先，把 i 加 1。因此，i 仍然小于或等于 n，因为根据归纳假设，在进入循环之前 $i < n$，而且 i 和 n 都是正整数。另外，把 *factorial* 设置成等于 $(i-1)! \cdot i = i!$，而根据归纳假设它是 $(i-1)!$。因此，p 仍然为真。所以 p 是循环不变量。换句话说，断言$[p \wedge (i < n)]\{S\}p$ 为真。由此得出 p {**while** $i < n$ S}$[(i \geq n) \wedge p]$也为真。

另外，在 $n-1$ 次执行之后循环终止而且满足 $i = n$，因为在程序开始时把 i 赋值成 1，每次执行都向 i 加 1，而当 $i \geq n$ 时循环终止。所以，在终止时 $factorial = n!$。■

下面将给出最后一个例子来说明如何用各种推理规则去验证较长的程序的正确性。

例 5　将要简述如何验证计算两个整数之积的程序 S 的正确性。

```
procedure multiply (m, n: 整数)
     { if n < 0 then a := - n
S1   { else a := n
     { k := 0
S2   { x := 0
     { while k < a
     { begin
S3   {     x := x + m
     {     k := k + 1
     { end
     { if n < 0 then product := - x
S4   { else product := x
```

目标是证明在执行 S 之后 $product$ 有值 mn。通过把 S 分成 $S = S_1;\ S_2;\ S_3;\ S_4$，如 S 的程序清单所示那样，就可以完成正确性证明。可以用合成规则来建立正确性证明。细节将留给读者作为练习。

设 p 是初始断言：m 和 n 都是整数。则可以证明当 q 是命题 $p \wedge (a = |n|)$ 时，$p\{S_1\}q$ 为真。下一步，设 r 是命题 $q \wedge (k = 0) \wedge (x = 0)$。容易验证 $q\{S_2\}r$ 为真。可以证明"$x = mk$ 而且 $k \leqslant a$"是 S_3 里的循环不变量。另外，容易看出，在 a 次循环之后循环终止而且 $k = a$，所以在这时 $x = ma$。因为 r 蕴涵着 $x = m \cdot 0$ 和 $0 \leqslant a$，所以在进入循环之前循环不变量为真。因为循环终止而且 $k = a$，由此得出 $r\{S_3\}S$ 为真，其中 S 是命题"$x = ma$ 而且 $a = |n|$。"最后，可以证明 S_4 是相对于初始断言 S 和终结断言 t 为正确的，其中 t 是命题"$product = mn$。"

把所有这些结果放到一起来考虑，因为 $p\{S_1\}q$，$q\{S_2\}r$，$r\{S_3\}S$，和 $S\{S_4\}t$ 都为真，所以从合成规则得出 $p\ \{S\}\ t$ 为真。另外，因为所有四个程序段都终止，所以 S 的确终止。这样就验证了这个程序的正确性。 ■

练习

1. 证明程序段

   $y := 1$

   $z := x + y$

   是相对于初始断言 $x = 0$ 和终结断言 $z = 1$ 为正确的。
2. 验证程序段

   if $x < 0$ **then** $x := 0$

   是相对于初始断言 **T** 和终结断言 $x \geqslant 0$ 为正确的。
3. 验证程序段

   $x := 2$

   $z := x + y$

   if $y > 0$ **then**

       $z := z + 1$

   else

   $z := 0$

   是相对于初始断言 $y = 3$ 和终结断言 $z = 6$ 为正确的。
4. 验证程序段

   if $x < y$ **then**

       $min := x$

   else

       $min := y$

   是相对于初始断言 **T** 和终结断言 $(x \leqslant y \wedge min = x) \vee (x > y \wedge min = y)$ 为正确的。

*5. 设计一条推理规则来验证形如

   if *condition* 1 **then**

```
        S1
    else if condition 2 then
        S2
        ⋮
    else
        Sn
```

的语句的部分正确性，其中 S_1，S_2，…，S_n 都是语句块。

6. 使用在练习 5 里建立的推理规则去验证程序

```
if x<0 then
    y := -2|x|/x
else if x>0 then
    y := 2|x|/x
else if x = 0 then
    y := 2
```

是相对于初始断言 T 和终结断言 $y=2$ 为正确的。

7. 用循环不变量证明下述计算实数 x 的 n 次方幂的程序是正确的，其中 n 是正整数。

```
power := 1
i := 1
while i≤n
begin
    power := power * x
    i := i + 1
end
```

*8. 证明在 3.4 节里给出的求 f_n 的迭代程序是正确的。

9. 给出在例 5 里给出的正确性证明的所有的细节。

10. 假定蕴涵式 $p_0 \to p_1$ 和程序断言 $p_1\{S\}q$ 都为真。证明 $p_0\{S\}q$ 也必然为真。

11. 假定程序断言 $p\{S\}q_0$ 和蕴涵式 $q_0 \to q_1$ 都为真。证明 $p\{S\}q_1$ 也必然为真。

12. 下面的程序计算商数和余数。

```
r := a
q := 0
while r≥d
begin
    r := r - d
    q := q + 1
end
```

验证它是相对于初始断言“a 和 d 都是正整数”和终结断言“q 和 r 是使得 $a = dq + r$ 和 $0 \leq r < d$ 的整数”为正确的。

13. 用循环不变量去验证欧几里德算法（2.4 节算法 1）是相对于初始断言“a 和 b 都是正整数”和终结断言“$x = \gcd(a, b)$”为部分正确的。

关键术语和结果

术语

定理：可以证明为真的数学断言

猜想：真值未知的数学断言

证明：对定理为真的说明

引理：用来证明其他定理的简单定理

推论：可以作为刚刚证明的定理的后果而被证明的命题

推理规则：用来从已知的断言得出结论的重言式蕴涵式

谬误：常常被不正确地用来得出结论而且是一种可能式的蕴涵式

循环推理或回避问题：其中一步或多步是基于被证明的命题的真值的推理

空证明：基于 p 为假的事实而对蕴涵式 $p\rightarrow q$ 为真的证明

平凡证明：基于 q 为真的事实而对蕴涵式 $p\rightarrow q$ 为真的证明

直接证明：通过证明当 p 为真时 q 必然为真而进行的对蕴涵式 $p\rightarrow q$ 为真的证明

间接证明：通过证明当 q 为假时 p 必然为假而进行的对蕴涵式 $p\rightarrow q$ 为真的证明

归谬证明：基于蕴涵式 $\neg p\rightarrow q$ 的真值（其中 q 是矛盾式）而对命题 p 为真的证明

分情形证明：证明每个前件都蕴涵结论因而对前件是命题的析取式的蕴涵式的证明

反例：使得 $P(x)$ 为假的元素 x

数学归纳法：包括基础步骤和归纳步骤而对形如 $\forall n\in \mathbf{N}P(n)$ 的命题的证明技术

基础步骤：在用数学归纳法证明 $\forall n\in \mathbf{N}P(n)$ 的过程里对 $P(1)$ 的证明

归纳步骤：在用数学归纳法证明 $\forall n\in \mathbf{N}P(n)$ 的过程里对 $P(n)\rightarrow P(n+1)$ 的证明

函数的递归定义：这样一种函数定义，其中规定一组初始的函数值，并规定从较小整数处的函数值获得较大整数处的函数值的规则

集合的递归定义：这样一种集合定义，其中规定集合里的一组初始元素，并规定从已知属于集合的元素获得其他元素的规则

递归算法：通过把问题归约到带有较小输入的同样问题而进行的算法

迭代：基于反复利用循环内的操作的过程

程序正确性：对过程总是产生正确结果的验证

循环不变量：在循环的每次执行期间都保持为真的性质

初始断言：规定程序的输入值所具有的性质的命题

终结断言：规定若程序正确地工作则输出值所应当具有的性质的命题

结果

良序性：非负整数的每个非空集合都有最小元

数学归纳法原理：若 $P(1)$ 为真而且 $\forall n[P(n)\rightarrow P(n+1)]$ 为真，则命题 $\forall nP(n)$ 为真

数学归纳法第二原理：若 $P(1)$ 为真而且 $\forall n[(P(1)\wedge\cdots\wedge P(n))\rightarrow P(n+1)]$ 为真，则命题 $\forall nP(n)$ 为真

复习题

1. a) 描述一下蕴涵式 $p\rightarrow q$ 的直接证明、间接证明和归谬证明分别是什么意思。

b）分别给出命题："若 n 是偶数，则 $n+4$ 是偶数"的直接证明、间接证明和归谬证明。

2. a）描述证明双向条件 $p \leftrightarrow q$ 的一种方式。

b）证明命题："整数 $3n+2$ 是奇数当且仅当整数 $9n+5$ 是偶数，其中 n 是整数。"

3. 为了证明命题 p_1，p_2，p_3 和 p_4 都是等价的，证明蕴涵式 $p_4 \to p_2$，$p_3 \to p_1$ 和 $p_1 \to p_2$ 都是有效的，这样做是否充分？若不充分，则给出可用来证明这四个命题都是等价的另外一组蕴涵式。

4. a）假定形如 $\forall x P(x)$ 的蕴涵式为假。如何可以证明它为假？

b）证明命题："对每个正整数 n 来说，数 n^2+1 是素数"为假。

5. a）构造性存在证明与非构造性存在证明之间的差异是什么？

b）证明："对每个整数 n 来说，都存在大于 n 的整数，它不能被 3 和 5 整除。你的存在性证明是构造性的还是非构造性的？

6. a）叙述一下正整数集合的良序性。

b）用该性质证明：每个正整数都可以写成素数之积。

7. a）你能否用数学归纳法原理来求出一个序列的前 n 项之和的公式？

b）你能否用数学归纳法原理来判定一个序列的前 n 项之和的给定公式是否正确？

c）求出前 n 个正偶数之和的公式，并且用数学归纳法证明它。

8. a）对哪些正整数 n 来说 $11n+17 \leqslant 2^n$ 为真？

b）用数学归纳法来证明你在 a）中所做的猜想。

9. a）仅用 5 分和 9 分的邮票，可以组成哪些数量的邮资？

b）用数学归纳法证明你所做的猜想。

c）用数学归纳法第二原理证明你所做的猜想。

d）找出与你在 b）和 c）里所给出的证明不同的对你的猜想的证明。

10. 给出使用数学归纳法第二原理的证明的三个不同的例子。

11. a）解释一下为什么若通过规定 $f(1)$ 以及从 $f(n-1)$ 求出 $f(n)$ 的规则来递归地定义一个函数，则这个函数是严格定义的。

b）给出函数 $f(n)=(n+1)!$ 的递归定义。

12. a）给出斐波那契数的递归定义。

b）证明：每当 $n \geqslant 3$ 时，就有 $f_n > \alpha^{n-2}$，其中 f_n 是斐波那契序列的第 n 项而 $\alpha=(1+\sqrt{5})/2$。

13. a）解释一下为什么若通过规定 a_1 和 a_2 以及对 $n=3, 4, 5, \cdots$ 来说从 a_1，a_2，…，a_{n-1} 求 a_n 的规则来递归地定义一个序列，则这个序列是严格定义的。

b）若 $a_1=1, a_2=2$，以及对 $n=3,4,5,\cdots$ 来说 $a_n=a_{n-1}+a_{n-2}+\cdots+a_1$，试求出 a_n 的值。

14. 给出两个例子说明对由元素和运算符组成的不同集合来说，如何递归地定义合式公式的。

15. a）给出字符串长度的递归定义。

b）用 a）的递归定义来证明 $l(xy)=l(x)+l(y)$。

16. a）什么是递归算法？

b）描述计算序列里 n 个数之和的递归算法。

17. 描述计算两个正整数的最大公因子的递归算法。

18. a) 测试一个计算机程序，看看对某些输入值来说它是否产生了正确的输出，是否这样就验证了这个程序总是产生正确的输出？
    b) 证明了一个计算机程序是相对于初始断言和终结断言为部分正确的，是否这样就验证了这个程序总是产生正确的输出？若不是，则还需要证明什么别的东西？
19. 你可以用什么技术来证明长的计算机程序是相对于初始断言和终结断言为部分正确的？
20. 什么是循环不变量？如何使用循环不变量？

补充练习

1. 证明：两个奇数之积还是奇数。
2. 证明：$\sqrt{5}$是无理数。
3. 证明或反驳：两个无理数之和还是无理数。
4. 证明或反驳：每当 n 是正整数时，n^2+n+1 就是素数。
5. 判定下面的论证是否有效：若 n 大于5，则 n^2 大于25。因此，若 n 是整数而且 n^2 大于25，则得出 n 大于5。
6. 证明当 n 不能被5整除时，n^4-1 能被5整除。使用分情形证明，分四种情形——当你把一个不能被5整除的整数除以5时，每一种非零余数对应一种情形。
7. 用分情形证明：$|xy|=|x||y|$。

*8. 把尤兰数定义成：$u_1=1$ 和 $u_2=2$。另外，在确定了小于 n 的整数是否尤兰数之后，若 n 可以唯一地写成两个不同的尤兰数之和，则让 n 等于下一个尤兰数。注意 $u_3=3$，$u_4=4$，$u_5=6$ 和 $u_6=8$。
   a) 求出前20个尤兰数。
   b) 证明存在无穷多个尤兰数。
9. 给出构造性的证明：存在多项式 $P(x)$ 使得 $P(x_1)=y_1, P(x_2)=y_2, \cdots, P(x_n)=y_n$，其中 $x_1,\cdots,x_n,y_1,\cdots,y_n$ 都是实数。

   [提示：设 $P(x)=\sum_{j=1}^{n}\left(\prod_{i\neq j}\frac{x-x_i}{x_j-x_i}\right)y_j$。]
10. 证明：每当 n 是正整数时，就有 $1^3+3^3+5^3+\cdots+(2n+1)^3=(n+1)^2(2n^2+4n+1)$。
11. 证明：每当 n 是正整数时，就有 $1\cdot2^0+2\cdot2^1+3\cdot2^2+\cdots+n\cdot2^{n-1}=(n-1)\cdot2^n+1$。
12. 证明：每当 n 是正整数时，就有

$$\frac{1}{1.3}+\frac{1}{3.5}+\cdots+\frac{1}{(2n-1)(2n+1)}=\frac{n}{2n+1}$$

13. 证明：每当 n 是正整数时，就有

$$\frac{1}{1.4}+\frac{1}{4.7}+\cdots+\frac{1}{(3n-2)(3n+1)}=\frac{n}{3n+1}$$

14. 证明：每当 n 是大于4的正整数时，就有 $2^n>n^2+n$。
15. 用数学归纳法证明：每当 n 是大于9的正整数时，就有 $2^n>n^3$。
16. 求出整数 N，使得每当 n 大于 N 时，就有 $2^n>n^4$。用数学归纳法证明你的结果是正确的。

17. 用数学归纳法证明：每当 n 是正整数时，就有 $a-b$ 是 a^n-b^n 的因子。

18. 用数学归纳法证明：每当 n 是非负整数时，就有 9 整除 $n^3+(n+1)^3+(n+2)^3$。

19. 算术级数是形如 $a, a+d, a+2d, \cdots, a+nd$ 的序列，其中 a 和 d 都是实数。用数学归纳法证明：$a+(a+d)+\cdots+(a+nd)=(n+1)(2a+nd)/2$ 给出了算术级数的项之和的公式。

20. 假定对 $j=1, 2, \cdots, n$ 来说 $a_j \equiv b_j \pmod{m}$。用数学归纳法证明：

    a) $\sum_{j=1}^{n} a_j \equiv \sum_{j=1}^{n} b_j \pmod{m}$

    b) $\prod_{j=1}^{n} a_j \equiv \prod_{j=1}^{n} b_j \pmod{m}$

*21. 确定哪些斐波那契数是偶数，用数学归纳法的一种形式来证明你的猜想。

*22. 确定哪些斐波那契数能被 3 整除，用数学归纳法的一种形式来证明你的猜想。

*23. 证明：对所有非负整数 n 来说 $f_k f_n + f_{k+1} f_{n+1} = f_{n+k+1}$，其中 k 是非负整数而且 f_i 表示第 i 个斐波那契数。

卢卡斯数的序列定义成：$l_0=2$，$l_1=1$，以及对 $n=2, 3, 4, \cdots$来说 $l_n = l_{n-1} + l_{n-2}$。

24. 证明：每当 n 是正整数时，就有 $f_n + f_{n+2} = l_{n+1}$，其中 f_i 和 l_i 分别是第 i 个斐波那契数和第 i 个卢卡斯数。

25. 证明：每当 n 是非负整数而且 l_i 是第 i 个卢卡斯数时，就有 $l_0^2 + l_1^2 + \cdots + l_n^2 = l_n l_{n+1} + 2$。

*26. 用数学归纳法证明：任意 n 个连续正整数之积能被 $n!$ 整除。[提示：利用恒等式 $m(m+1)\cdots(m+n-1)/n! = (m-1)m(m+1)\cdots(m+n-2)/n! + m(m+1)\cdots(m+n-2)/(n-1)!$。]

27. 用数学归纳法证明：每当 n 是正整数时，就有 $(\cos x + i\sin x)^n = \cos nx + i\sin nx$。[提示：利用恒等式 $\cos(a+b) = \cos a \cos b - \sin a \sin b$ 和 $\sin(a+b) = \sin a \cos b + \cos a + \sin b$。]

*28. 用数学归纳法证明：每当 n 是正整数而且$\sin(x/2)\neq 0$ 时，就有 $\sum_{j=1}^{n} \cos jx = \cos[(n+1)x/2]\sin(nx/2)\sin(x/2)$。

麦卡锡 91 函数定义成：对所有正整数 n 来说应用规则

$$M(n)=\begin{cases} n-10 & 若\ n>100 \\ M(M(n+11)), & 若\ n\leqslant 100 \end{cases}$$

29. 通过连续地使用$M(n)$的定义规则来求出

    a) $M(102)$ b) $M(101)$ c) $M(99)$ d) $M(97)$ e) $M(87)$ f) $M(76)$

**30. 证明：函数 $M(n)$ 是严格定义的从正整数集合到正整数集合的函数。[提示：证明对所有满足 $n\leqslant 101$ 的正整数 n 来说都有$M(n)=91$。]

31. 每当 n 是正整数时就有

$$\frac{1}{1.2}+\frac{1}{2.3}+\cdots+\frac{1}{(n-1)\ n}=\frac{3}{2}-\frac{1}{n}$$

下面给出的证明是否正确？为你的答案给出理由。

基础步骤：当 $n=1$ 时结果为真，因为

$$\frac{1}{1.2}=\frac{3}{2}-\frac{1}{1}$$

归纳步骤：假定对 n 来说结果为真，则

$$\frac{1}{1.2}+\frac{1}{2.3}+\cdots+\frac{1}{(n-1)n}+\frac{1}{n(n+1)}=\frac{3}{2}-\frac{1}{n}+(\frac{1}{n}-\frac{1}{n+1})=\frac{3}{2}-\frac{1}{n+1}$$

因此，若对 n 来说结果为真，则对 $n+1$ 来说结果为真。证毕。

*32. 一种拼图玩具是通过把可以拼在一起的图片合并成块来完成的。每次当把一个图片加入一个块时，或者当把两个块合并时，就算做一步。用数学归纳法的第二种形式证明：无论如何完成各步，组装含有 n 个图片的拼图玩具都恰好需要 $n-1$ 步。

*33. 证明：若在 n 个圆中每两个都恰好相交于两点，而任意三个圆都没有公共点，则这些圆把平面划分成 n^2-n+2 个区域。

*34. 证明：若在 n 个平面中任意三个都有公共点，而任意四个都没有公共点，则这些平面把三维空间划分成 $(n^3+5n+6)/6$ 个区域。

*35. 用良序性证明：$\sqrt{2}$是无理数。[提示：假定$\sqrt{2}$是有理数。证明形如 $b\sqrt{2}$的正整数组成的集合有最小元 a。然后证明 $a\sqrt{2}-a$ 是具有这种形式的更小的正整数。]

*36. 若一个集合的每个非空子集合都有最小元，则这个集合是良序的。判断下面的每个集合是否良序的。

a）整数集合

b）大于 -100 的整数的集合

c）正有理数集合

d）分母小于 100 的正有理数的集合

*37. 证明：若把数字归纳法原理作为一条公理，则可以证明良序性。

*38. 证明：数学归纳法的第一和第二原理是等价的；即从一个原理是有效的就可以证明另外一个原理是有效的。

39. a）证明：若 $a_1,a_2,\cdots,a_n$ 都是正整数，则 $\gcd(a_1,a_2,\cdots,a_{n-1},a_n)=\gcd(a_1,a_2,\cdots,a_{n-2},\gcd(a_{n-1},a_n))$。

b）利用 a）和欧几里德算法来建立一个递归算法，计算 n 个正整数的最大公因子。

*40. 描述一个递归算法，把 n 个正整数的最大公因子写成这些整数的线性组合。

*41. 求出 $f(n)$ 的显式公式，其中 $f(1)=1$，而且若 $n\geqslant 2$，则 $f(n)=f(n-1)+2n-1$。用数学归纳法证明你的结果。

**42. 给出由所含有的 0 是 1 的两倍的位串所组成的集合的递归定义。

43. 设 S 是位串的集合，它递归地定义成：$\lambda\in S$，以及若 $x\in S$，则 $0x\in S$，$x1\in S$，其中 λ 是空串。

a）求出 S 中所有长度不超过 5 的串。

b）给出对 S 中元素的显式描述。

44. 设 S 是字符串的集合，它递归地定义成 $abc\in S$，$bac\in S$，$acb\in S$，以及若 $x\in S$ 则

$abcx \in S$，$abxc \in S$，$axbc \in S$ 和 $xabc \in S$。

a）求出 S 中长度为 8 或更短的所有串。

b）证明：S 中每个元素都有能被 3 整除的长度。

由所有平衡的括号串组成的集合递归地定义成：$\lambda \in B$，其中 λ 是空串；若 x，$y \in B$，则 $(x) \in B$，$xy \in B$。

45. 求出所有带 4 个或更少符号的平衡的括号串。

46. 用归纳法证明：若 x 是平衡的括号串，则在 x 里左括号的个数等于右括号的个数。在括号串的集合上定义函数 N：

$$N(\lambda)=0, \quad N(()=1, N())=-1,$$
$$N(uy)=N(u)+N(v),$$

其中 λ 是空串，u 和 v 都是串。可以证明 N 是严格定义的。

47. 求出

a) $N(())$　　b) $N()))())()$　　c) $N((()(()$　　d) $N(()((()))(()))$

**48. 证明：括号串 w 是平衡的，当且仅当 $N(w)=0$，而且每当 u 是 w 的前缀（即 $w=uv$）时，就有 $N(u) \geqslant 0$。

*49. 给出一个求所有包含 n 个或更少符号的平衡的括号串的递归算法。

50. 若 $a=b$，则 $\gcd(a, b)=a$；若 a 和 b 都是偶数，则 $\gcd(a,b)=2\gcd(a/2,b/2)$；若 a 是偶数而 b 是奇数，则 $\gcd(a,b)=(a/2,b)$；若 a 和 b 都是奇数则 $\gcd(a,b)=\gcd(b-a,b)$。根据上述事实，给出一个求满足 $a \leqslant b$ 的两个非负整数 a 和 b 的最大公因子的递归算法。

51. 验证程序段

```
if x > y then
    x := y
```

是相对于初始断言 T 和终结断言 $x \leqslant y$ 为正确的。

*52. 建立一条验证递归程序的推理规则，并用它去验证在 3.4 节给出的计算阶乘的递归程序。

计算机题目

编写带有下述输入与输出的程序。

1. 给定几何级数 a，ar，ar^2，…，ar^n，求出它的各项之和。

2. 给定非负整数 n，求出 n 个最小的正整数之和。

**3. 给定去掉一个 $2^n \times 2^n$ 格子的棋盘，用 L 形状的拼片构造这个棋盘的铺砖系统。

**4. 对含有变量 x，y 和 z 以及运算符 $\{+, *, /, -\}$ 的表达式来说，生成所有带有 n 个或更少符号的合式公式。

5. 生成所有带有 n 个或更少符号的命题的合式公式，其中每个符号是 T、F、命题变量 p 和 q 之一或 $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$ 里的一个运算符。

6. 给定一个字符串，求出它的倒置。

7. 给定实数 a 和非负整数 n，用递归求 a^n。

8. 给定实数 a 和非负整数 n，用递归求 a^{2^n}。

*9. 给定实数 a 和非负整数 n，利用 n 的二进制展开式和计算 a^{2^k} 的递归算法来求 a^n。

10. 给定两个不全为零的整数，用递归求它们的最大公因子。

11. 给定整数的列表和元素 x，用线性搜索的递归实现求 x 在这个列表中的位置。

12. 给定整数的列表和元素 x，用二叉搜索的递归实现求 x 在这个列表中的位置。

13. 给定非负整数 n，用迭代来求第 n 个斐波那契数。

14. 给定非负整数 n，用递归来求第 n 个斐波那契数。

15. 给定一个正整数，求出这个整数的划分的数目。(参见 3.3 节练习 35。)

16. 给定正整数 m 和 n，求出阿克曼函数在 (m,n) 处的值 $A(m,n)$。(参见 3.3 节练习 36 前面的说明。)

计算和研究

用计算程序或你已经编写的程序做下面的习题。

1. 对 $n \leqslant 10000$ 验证哥德巴赫猜想，它断言每个正偶数 n 都是两个素数之和。

2. 对满足 $n \leqslant 20$ 的所有正整数 n 求出 $n!+1$ 的最小素因子。

3. 对满足 $n \leqslant 10$ 的每个正整数 n 求出最小的 n 个连续的合数。

4. 一个古老而未解决的猜想说存在无穷多对孪生素数，即相差为二的素数。你能找出多少对孪生素数？

5. 确定哪些斐波那契数能被 5 整除、哪些能被 7 整除、哪些能被 11 整除。证明你的猜想是正确的。

6. 用 L 形状的拼片对去掉一个格子的 16×16，32×32 和 64×64 的棋盘构造铺砖系统。

7. 探索用 L 形状的拼片可以完全地覆盖哪些棋盘。

8. 著名的难题 $3x+1$ 猜想（也称为科拉兹猜想以及许多其他的名字）说，无论你从哪个整数 x 开始，对函数 $f(x)$ 进行迭代，其中若 x 是偶数则 $f(x)=x/2$，若 x 是奇数则 $f(x)=3x+1$，结果总是产生整数 1。对尽可能多的正整数验证这个猜想。

9. 阿克曼函数的哪些值是足够小的，使得你能够计算它们？

10. 比较一下递归地计算斐波那契数与迭代地计算它们所需要的运算次数或时间。

写作题目

利用本书以外的资料，针对下面的内容写出短文。

1. 描述一下数学归纳法的起源。哪些是使用它的第一批人？他们把它用在哪些问题上？

2. 在过去的 20 年里，基于大量的计算机计算，已经证明了若干重要的定理。讨论一下这样的证明的有效性，并且描述围绕基于计算机计算的证明的争论。

3. 逻辑编程利用推理规则，在用量词、谓词和逻辑联结词表示的命题上操作。解释一下逻辑编程的基本概念和如何在人工智能里利用它。用编程语言 PROLOG 说明它的用法。

4. “自动定理证明”是用计算机来机械地证明定理的任务。讨论一下自动定理证明的目标和

应用，以及在发展自动定理证明器上所取得的进步。

5. 描述一下雷曼·艾伦开发的现代逻辑游戏WFF'N PROOF 的规则（因为它是在 20 世纪 60 年代中期出版的，你也许不得不从旧物品里找一套）。给出在WFF'N PROOF 里包含的游戏的一些例子。
6. 在 3.2 节的练习里使用的 L 形状的拼片，是哥伦布在 1954 年引入的多元多米诺骨牌的一些例子。描述一下关于用多元多米诺骨牌铺满棋盘的一些问题和有关的结果。
7. 描述一下在递归定义的理论中以及在集合并集算法的复杂性分析中，对阿克曼函数的使用情况。
8. 描述一下在刘易斯·卡洛尔的著作中所找到的某些逻辑问题，并且说明一下如何使用推理规则去解决这些问题。
9. 讨论一下用来证明程序正确性的各种方法中的一些方法，并且把它们与 3.5 节描述的霍尔方法进行比较。
10. 解释一下如何扩充程序正确性的想法和概念，去证明操作系统是安全的。

第4章　计　　数

组合数学这一研究个体安排的学科，是离散数学的重要部分。早在17世纪就开始了这类课题的研究，当时在赌博游戏的研究中出现了组合问题。枚举，具有确定性质的个体的计数，是组合数学的一个重要的部分。我们必须对个体计数来求解许多不同类型的问题。例如，用计数确定算法的复杂性。计数也用于确定是否存在着能够充分满足需求的电话号码或因特网址。此外，计数技术也广泛用于计算事件的概率。

4.1节将要研究的基本计数规则可以求解各种各样的问题。例如，我们可以用这些规则来计数美国各种不同可能的电话号码，计算机系统中允许使用的登录密码，以及在比赛结束时赛跑运动员的不同的名次。另一个重要的组合工具是鸽巢原理，我们将放在4.2节研究。这个原理指出，当把物体放在盒子里时，若物体比盒子多，那么存在一个盒子包含着至少两个物体。例如，我们可以用这个原理证明在15个或者更多的学生中至少有3人出生在相同的星期几。

我们可以把许多计数问题用集合中个体的有序或无序安排来描述。这些安排称作排列和组合，在许多的计数问题中都会用到它们。例如，在2000个学生参加的考试竞赛中最终将有100个获胜者被邀请赴宴。我们可以枚举将被邀请的100个学生的可能的组合，以及最终10名获奖者的产生方式。

我们可以使用计数技术分析赌博游戏，如扑克。我们也可以使用这些技术来确定抽奖获胜的概率，例如一个人从前48个正整数中选取6个数中奖的概率。

组合数学的另一个问题涉及到生成某个特定类型的所有排列。这在计算机模拟中通常是很重要的。我们将设计算法来生成各种类型的排列。

4.1　计数的基础

4.1.1　引言

一个计算机系统的登录密码由6，7或8个字符组成。每个字符必须是数字或字母表中的字母。每个密码必须至少包含一位数字，问有多少个密码？回答这个问题及各种各样的其他计数问题所需要的技术将在这一节引入。

整个数学和计算机科学中存在着计数问题。例如，我们必须计数成功的实验结果和所有可能的实验结果以确定离散事件的概率。我们需要计数某个算法用到的操作数来研究它的时间复杂性。

在这一节我们将引入基本的计数技术。这些方法是几乎所有计数技术的基础。

4.1.2　基本的计数原则

我们将提出两个基本的计数原则。然后我们将说明怎样用它们来求解许多不同的计数问题。

求和法则　如果完成第一项任务有 n_1 种方式，第二项任务有 n_2 种方式，并且这两项任务不能同时完成，那么完成第一项或第二项任务有 n_1+n_2 种方式。

下面的例子说明怎样使用求和法则。

例 1　假定要选一位数学教师或数学专业的学生作为校委会的代表。如果有 37 位数学教师和 83 位数学专业的学生，那么这个代表有多少种不同的选择？

解　完成第一项任务，选一位数学教师，可以有 37 种方式。完成第二项任务，选一位数学专业的学生，有 83 种方式。根据求和法则，结果有 37 + 83 = 120 种可能的方式来挑选这个代表。■

我们可以把求和法则推广到多于两项任务的情况。假定任务 T_1，T_2，…，T_m 分别有 n_1，n_2，…，n_m 种完成的方式，并且任何两项任务都不能同时做，那么完成其中一项任务的方式数是 $n_1+n_2+\cdots+n_m$。正如例 2 和例 3 所示，这个推广的求和法则在计数问题中是常常用到的。这个求和法则可以从两个集合的求和法则使用数学归纳法加以证明（见节后的练习 53）。

例 2　一个学生可以从三个表中的一个表选择一个计算机课题。这三个表分别包含 23，15 和 19 个可能的课题。那么被选择的课题可能有多少种？

解　这个学生有 23 种方式从第一个表中选择课题，有 15 种方式从第二个表中选择课题，有 19 种方式从第三个表中选择课题。因此，共有 23 + 15 + 19 = 57 种选择课题的方式。■

例 3　在下面的代码被执行后 k 的值是什么？

```
k := 0
for i1 := 1 to n1
  k := k + 1
for i2 := 1 to n2
  k := k + 1
  ⋮
for im := 1 to nm
  k := k + 1
```

解　k 的初值是 0。这个代码块由 m 个不同的循环构成。循环中的每次执行，k 都要加 1。令 T_i 是执行第 i 个循环的任务。因为第 i 个循环被执行 n_i 次，所以任务 T_i 可以用 n_i 种方式完成。由于任何两个任务不能同时执行，求和法则证明 k 的最后值，即完成任务 $T_i(i=1,2,\cdots,m)$ 的方式数是 $n_1+n_2+\cdots+n_m$。■

求和法则可以用集合的语言表述如下：如果 A_1，A_2，…，A_m 是不交的集合，那么在其并集中的元素数是每个集合的元素数之和。为把这种表述与求和法则联系起来，令 T_i 是从 A_i $(i=1,2,\cdots,m)$ 中选取一个元素的任务。有 $|A_i|$ 种方式做 T_i。由于任何两个任务不可能同时做，根据求和法则，从其中某个集合选择一个元素的方式数，即在并集中的元素数是

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|$$

这个等式仅适用于问题中的集合是不相交的情况。当这些集合含有公共元素时情况要复杂得多。本节的后面将对这种情况进行简要的讨论，更深入的讨论放在第 5 章。

当一个过程是由独立的任务组成时适用于乘积法则。

乘积法则 假定一个过程可以被分解成两个任务。如果完成第一个任务有 n_1 种方式，在第一个任务完成之后有 n_2 种方式完成第二个任务，那么完成这个过程有 n_1n_2 种方式。

下面的例子显示怎样来使用乘积法则。

例 4 用一个字母和一个不超过 100 的正整数给礼堂的座位编号。那么不同编号的座位最多有多少？

解 给一个座位编号的过程由两个任务组成，即从 26 个字母中先选择一个字母分配给这个座位，然后再从 100 以内的正整数中选择一个整数分配给它。乘积法则表明一个座位可以有 $26 \cdot 100 = 2600$ 种不同的编号方式。因此，不同编号的座位数至多是 2600。 ■

例 5 某个计算机中心有 32 台微机，每台微机有 24 个端口。问在这个中心里有多少个不同的单机端口？

解 选择一个端口的过程由两个任务组成。首先挑一台微机，然后在这台微机上挑一个端口。因为有 32 种方式选择微机，而不管选择了哪台微机，又有 24 种方式选择端口，由乘积法则表明有 768 个端口。 ■

经常会用到推广的乘积法则。假定一个过程由执行任务 T_1，T_2，…，T_m 来完成。如果在完成任务 T_1，T_2，…，T_{i-1}之后用 n_i 种方式来完成 T_i，那么完成这个过程有 $n_1 \cdot n_2 \cdot \cdots \cdot n_m$ 种方式。可以由两个任务的乘积法则通过数学归纳法证明推广的乘积法则（见节后的练习 54）。

例 6 有多少个不同的 7 位二进制串？

解 每位可以是 0 或 1，有两种选择方式。因此，乘积法则表明总共有 $2^7 = 128$ 个不同的 7 位二进制串。 ■

例 7 如果每个车牌由 3 个字母后跟 3 个数字的序列构成（任何字母的序列都允许，即使它们是令人讨厌的），那么有多少个不同的有效的车牌？

解 对 3 个字母中的每个字母有 26 种选择，对 3 数字中的每个数字有 10 种选择。因此，由乘积法则总共有 $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17\,576\,000$ 个可能的车牌。 ■

例 8 计数函数。从一个 n 元集到一个 m 元集存在多少个函数？

解 一个函数对于定义域中 m 个元素中的每个元素都要选择陪域中 n 个元素中的一个元素来对应。因此，由乘积法则存在 $n \cdot n \cdot \cdots \cdot n = n^m$ 个从 m 元集到 n 元集的函数。 ■

例 9 一对一函数。从一个 m 元素集合到一个 n 元素集合存在多少个一对一函数？

解 首先注意到当 $m > n$ 时没有从 m 元集到 n 元素集合的一对一函数。现在令 $m \leqslant n$。假设定义域中的元素是 a_1，a_2，…，a_m。有 n 种方式选择函数在 a_1 的值。因为函数是一

对一的，可以有 $n-1$ 种方式选择函数在 a_2 的值（因为被 a_1 用过的值不能再用）。一般地，有 $n-k+1$ 种方式选择函数在 a_k 的值。由乘积法则，从一个 m 元集到一个 n 元集存在着 $n(n-1)(n-2)\cdots(n-m+1)$ 个一对一函数。■

例 10 电话编号计划。在北美电话号码的格式是由一个编号计划规定的。一个电话号码由 10 个数字组成，这些数字有一个 3 位的地区代码，一个 3 位的局代码，以及一个 4 位的话机代码。出于信号的考虑，在一些数字上有某种限制。为了规定允许的格式，令 X 表示可以在 0 到 9 之间任意选取的数字，N 表示可以在 2 到 9 之间选取的数字，而 Y 表示必须取 0 或 1 的数字。下面讨论两个编号计划，分别称为老计划和新计划（老计划是 20 世纪 60 年代使用的，已经被新计划代替了，但目前对新号码需求的迅速增长甚至使得这个新计划也将显得陈旧了）。正如将要证明的，新计划允许使用更多的号码。

在老计划中，地区代码、局代码和话机代码的格式分别为 NYX，NNX 和 XXXX。在新计划中，这些代码的格式分别为 NXX，NXX 和 XXXX。在老计划和新计划下可能有多少个不同的北美电话号码？

解 由乘积法则，格式为 NYX 的地区代码有 $8\cdot2\cdot10=160$ 个，格式为 NXX 的地区代码有 $8\cdot10\cdot10=800$ 个。类似地，由乘积法则，存在 $8\cdot8\cdot10=640$ 个格式为 NNX 的局代码和 $8\cdot10\cdot10=800$ 个格式为 NXX 的局代码。乘积法则也表明存在着 $10\cdot10\cdot10\cdot10=10\ 000$ 个格式为 XXXX 的话机代码。

因此，再次使用乘积法则，结果在老计划下存在

$$160\cdot640.10\ 000=1\ 024\ 000\ 000$$

个不同的北美有效的电话号码。在新计划下存在

$$800\cdot800\cdot10\ 000=6\ 400\ 000\ 000$$

个不同的电话号码。■

例 11 执行下面的代码以后 k 的值是什么？

```
k := 0
for i1 := 1 to n1
  for i2 := 1 to n2
      ⋮
  for im := 1 to nm
    k := k + 1
```

解 k 的初值是 0。这个嵌套的循环每执行一次，k 就加 1。令 T_i 表示执行第 i 个循环的任务，那么循环执行的次数就是完成任务 T_1，T_2，…，T_m 的方法数。因为对每个整数 i_j，$1\leqslant i_j\leqslant n_j$，第 j 个循环都执行一次，执行任务 T_j，$j=1,2,\cdots,m$，的方法数就是 n_j。由乘积法则，这个嵌套的循环执行了 $n_1n_2\cdots n_m$ 次。因此 k 最后的值是 $n_1n_2\cdots n_m$。■

例 12 计数有穷集的子集。用乘积法则证明一个有穷集 S 的不同的子集数是 $2^{|S|}$。

解 设 S 是有穷集。按任意的顺序将 S 的元素列成一个表。考虑到在 S 的子集和长为 $|S|$ 的二进制串之间存在着一对一的对应，即如果表的第 i 个元素在这个子集里则该子集对

应的二进制串的第 i 位为 1，否则这位为 0。由乘积法则，存在着 $2^{|S|}$ 个长为 $|S|$ 的二进制串。因此 $|P(S)|=2^{|S|}$。■

乘积法则也常用集合的语言表述如下：如果 A_1，A_2，…，A_m 是有穷集，那么在这些集合的笛卡儿积中的元素数是每个集合的元素数之积。为把这种表述与乘积法则联系起来，注意到在笛卡儿积 $A_1\times A_2\times\cdots\times A_m$ 中选一个元素的任务是通过在 A_1 中选一个元素，A_2 中选一个元素，…，A_m 中选一个元素来完成的。由乘积法则得到

$$|A_1\times A_2\times\cdots\times A_m|=|A_1|\cdot|A_2|\cdot\cdots\cdot|A_m|$$

比较复杂的计数问题 许多计数问题不能仅仅使用求和法则或者乘积法则来求解。但是，许多复杂的计数问题可以使用这两个法则来求解。

例 13 在计算机语言 BASIC 的某个版本中，变量的名字是含有一个或两个字符的符号串，其中的大写和小写字母是不加区分的（一个字符或者取自 26 个英文字母，或者取自 10 个数字）。此外，变量名必须以字母作为开始，并且必须和由两个字符构成的用于程序设计的 5 个保留字相区别。在 BASIC 的这个版本中有多少个不同的变量名？

解 令 V 等于在这个 BASIC 版本中的不同的变量名个数，V_1 是单字符的变量名个数，V_2 是两个字符的变量名个数。那么由求和法则，$V=V_1+V_2$。由于单字符变量名必须是字母，故 $V_1=26$。又根据乘积法则存在 $26\cdot 36$ 个以字母打头以字母数字结尾的 2 位字符串。但是其中 5 个不包含在内，因此 $V_2=26\cdot 36-5=931$。从而在这个 BASIC 版本中存在 $V=V_1+V_2=26+931=957$ 个不同的变量名。■

例 14 计算机系统的每个用户有一个 6 到 8 个字符构成的登录密码，其中每个字符是一个大写字母或者数字，且每个密码必须至少包含一个数字。有多少可能的密码？

解 令 P 是可能的密码总数，且 P_6，P_7，P_8 分别表示 6，7 或 8 位的可能的密码数。由求和法则，$P=P_6+P_7+P_8$。我们现在找 P_6，P_7 和 P_8。直接找 P_6 是困难的。而找 6 个大写字母和数字构成的字符串数是容易的，其中包含那些没有数字的串在内，然后从中减去没有数字的串数就得到 P_6。由乘积法则，6 个字符的串数是 36^6，而没有数字的字符串数是 26^6。因此，

$$P_6=36^6-26^6=2\ 176\ 782\ 336-308\ 915\ 776\ =1\ 867\ 866\ 560$$

类似地得到

$$P_7=36^7-26^7=78\ 364\ 164\ 096\ -8\ 031\ 810\ 176\ =70\ 332\ 353\ 920$$

和

$$P_8=36^8-26^8=2\ 812\ 109\ 907\ 456\ -208\ 827\ 064\ 576=2\ 612\ 282\ 842\ 880$$

从而，

$$P=P_6+P_7+P_8=2\ 684\ 483\ 063\ 360$$ ■

例 15 计数因特网网址。在由计算机的物理网络互连而构成的因特网中，每台计算机（或者更精确地说是计算机的每个网络连接）被分配一个因特网地址。目前正在使用的网际协议版本 4（IPv4）中，一个地址是一个 32 位的二进制串。它以网络标识（netid）开始，后面跟随着主机标识（hostid），该标识把一个计算机认定为某个指定网络的成员。

根据网络标识和主机标识位数的不同使用 3 种地址形式。用于最大网络的 A 类地址，由 0 后跟 7 位的网络标识和 24 位的主机标识构成。用于中等规模网络的 B 类地址，由 10 后跟 14 位的网络标识和 16 位的主机标识构成。用于最小网络的 C 类地址，由 110 后跟 21 位的网络标识和 8 位的主机标识构成。由于特定用途对地址有着某些限制：1111111 在 A 类网络的网络标识中是无效的，全 0 和全 1 组成的主机标识对任何网络都是无效的。因特网上的一台计算机有一个 A 类、B 类或 C 类地址。（除了 A 类、B 类和 C 类地址外，还有 D 类地址和 E 类地址。D 类地址在多台计算机同时编址时用于多路广播，它由 1110 后跟 28 位组成。E 类地址为将来应用，由 11110 后跟 27 位组成。不会把 D 和 E 类地址分配给因特网中的一台计算机作为 IP 地址。）图 4-1 显示了 IPv4 的编址。（A 类和 B 类网络标识的数量限制已经使得 IPv4 编址不够用了；将代替 IPv4 的 IPv6 使用 128 位地址来解决这个问题。）

对因特网上的计算机有多少不同的有效 IPv4 地址？

位数	0	1	2	3	4	8	16	24	31
A类	0	网络标识					主机标识		
B类	1	0	网络标识					主机标识	
C类	1	1	0	网络标识					主机标识
D类	1	1	1	0	多路广播地址				
E类	1	1	1	1	0	地址			

图 4-1 因特网地址（IPv4）

解 令 x 是因特网上计算机的有效地址数，X_A，X_B 和 X_C 分别表示 A 类、B 类和 C 类的有效地址数。由求和法则，$X = X_A + X_B + X_C$。

为了找到 X_A，由于 1111111 是无效的，故存在 $2^7 - 1 = 127$ 个 A 类的网络标识。对于每个网络标识，存在 $2^{24} - 2 = 16\ 777\ 214$ 个主机标识，这是由于全 0 和全 1 组成的主机标识是无效的。因此，$X_A = 127 \cdot 16\ 777\ 214 = 2\ 130\ 706\ 178$。

为了找到 X_B 和 X_C，首先注意到存在 $2^{14} = 16\ 384$ 个 B 类网络标识和 $2^{12} = 2\ 097\ 152$ 个 C 类网络标识。对每个 B 类网络标识存在着 $2^{16} - 2 = 65\ 534$ 个主机标识，而对每个 C 类网络标识存在着 $2^8 - 2 = 254$ 个主机标识，这也是考虑到全 0 和全 1 组成的主机标识是无效的。因而，$X_B = 1\ 073\ 709\ 056$，$X_C = 532\ 676\ 608$。我们可以断言 IPv4 有效地址的总数是 $X = X_A + X_B + X_C = 2\ 130\ 706\ 178 + 1\ 073\ 709\ 056 + 532\ 676\ 608 = 3\ 737\ 091\ 842$。■

4.1.3 容斥原理

当同时做两个任务时，我们不能使用求和法则来计数完成其中一个任务的方式。把对每个任务的方式数加起来将导致计数结果的增大，因为完成公共任务的方式被计数了两次。为了正确地计数完成其中一个任务的方式，我们先把完成每个任务的方式数加起来，然后再减去完成公共任务的方式数。这个技术叫做*容斥原理*。例 16 显示了我们可以怎样用这个原理来求解计数问题。

例 16　以 1 开始或者以 00 结束的 8 位二进制符号串有多少个？

解　第一个任务，构造以 1 开始的 8 位二进制字符串，完成它有 $2^7=128$ 种方式，这是由乘积法则得到的。因为第一位只有一种选择方式，而其他 7 位中的每位有 2 种选择方式。

第二个任务，构造以 00 结束的 8 位二进制字符串，完成它有 $2^6=64$ 种方式，这也是由乘积法则得到的。因为前 6 位的每位有 2 种选择方式，而最后两位只有一种选择方式。

同时完成两个任务，构造以 1 开始以 00 结束的 8 位二进制符号串，完成它有 $2^5=32$ 种方式。这里也使用了乘积法则，因为第一位只有一种选法，最后两位也只有一种选法。因而，以 1 开始或者以 00 结束的 8 位二进制符号串个数，即完成第一或第二个任务的方式数，等于 $128+64-32=160$。■

我们可以用集合的语言表述这一计数原理。令 A_1 和 A_2 是集合，T_1 是从 A_1 选择一个元素的任务，T_2 是从 A_2 选择一个元素的任务。完成 T_1 有 $|A_1|$ 种方式，完成 T_2 有 $|A_2|$ 种方式。完成 T_1 或 T_2 的方式数是完成 T_1 的方式数与完成 T_2 的方式数之和减去同时完成 T_1，T_2 两个任务的方式数。因为存在 $|A_1\cup A_2|$ 种方式完成 T_1 或 T_2，$|A_1\cap A_2|$ 种方式同时完成 T_1 和 T_2，我们有

$$|A_1\cup A_2|=|A_1|+|A_2|-|A_1\cap A_2|$$

这就是在 1.5 节给出的计数两个集合并集中元素的公式。

容斥原理可以推广来求完成 n 个不同任务中的一个任务的方式数，或者换一种说法，就是找 n 个集合的并集中的元素数，其中 n 是正整数。我们将在第 5 章研究这个容斥原理和它的某些广泛应用。

4.1.4　树图

可以使用树图求解计数问题。一棵树由根、从根出发的许多分支以及可能从其他分支端点出发的新的分支构成（我们将在第 8 章详细地研究树）。为了在计数中使用树，我们用一个分支表示每个可能的选择，用树叶表示可能的结果。这些树叶是某些分支的端点，从这些端点不再进一步分支。

例 17　有多少不含连续两个 1 的 4 位二进制串？

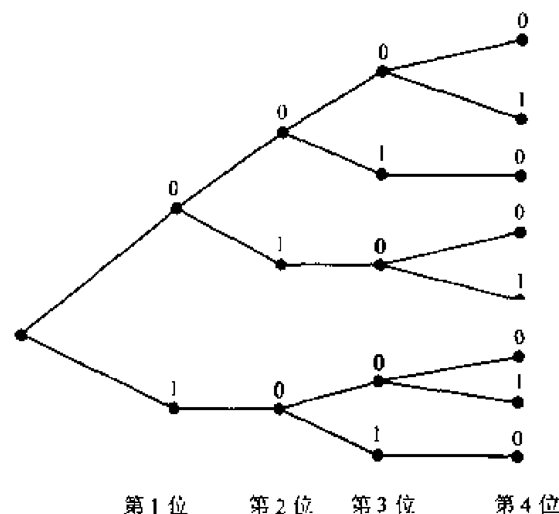


图 4-2　不含连续两个 1 的 4 位二进制串

解　图 4-2 的树图给出了所有不含连续两个 1 的 4 位二进制串。我们看出存在 8 个不含连续两个 1 的 4 位二进制串。■

例 18　在两个队（队$_1$ 和队$_2$）之间的决赛至多由 5 次比赛构成。先胜 3 次的队赢得决赛。决赛可能出现多少种不同的方式？

解　在图 4-3 的树图中以每次比赛的得胜者给出了决赛可以进行的所有方式。我们看到有 20 种不同的决赛的方式。■

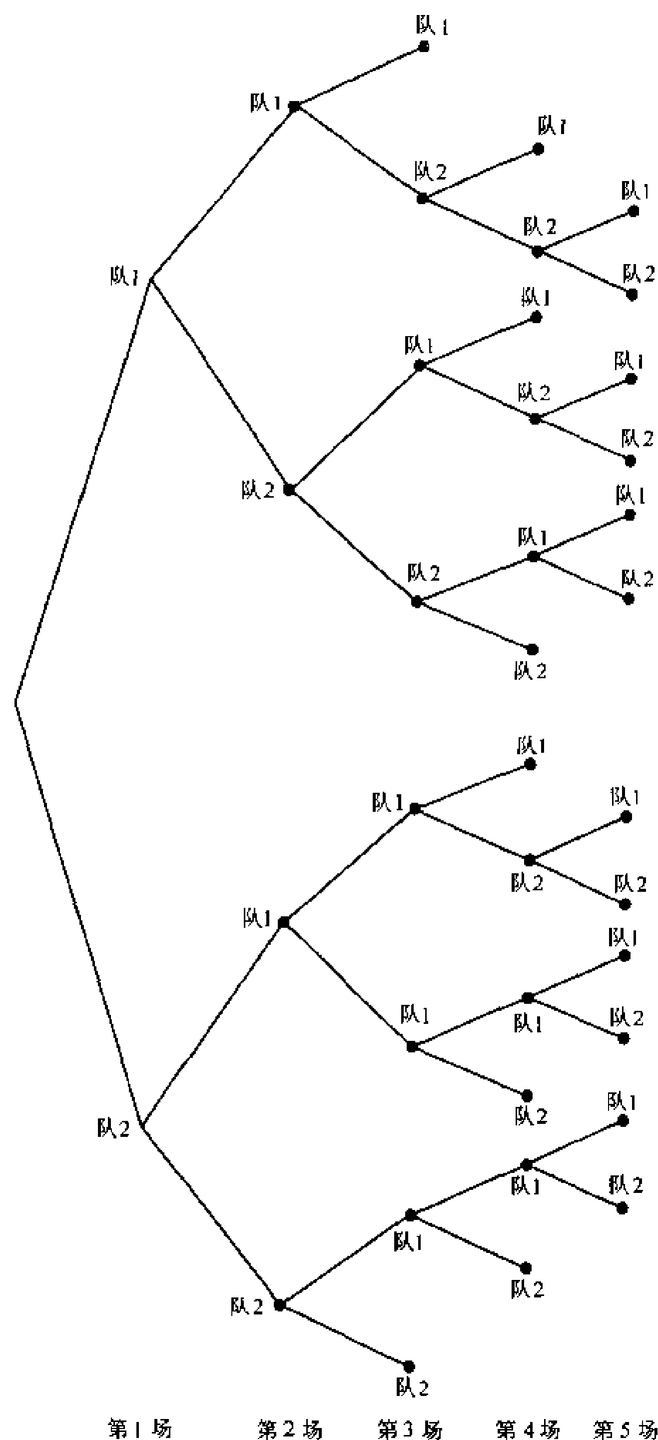


图 4-3　5 次决赛胜 3 次

练习

1. 一个学院有 18 个数学专业的和 325 个计算机科学专业的学生。
   a）选两个代表，使得一个是数学专业的且另一个是计算机科学专业的，有多少种方式？
   b）选一个数学专业或计算机科学专业的代表又有多少种方式？
2. 一个办公大楼有 27 层，并且每层有 37 个办公室，那么在这个大楼里有多少个办公室？

3. 一次多项选择考试包含 10 个问题。每个问题有 4 个可能的答案。
   a）在这次考试中如果每个问题都要回答，一个学生回答这些问题可能有多少种方式？
   b）在这次考试中如果允许某些答案空缺，一个学生回答这些问题可能有多少种方式？
4. 某种商标的衬衫有 12 种颜色，有男式和女式 2 种样式，每种样式有 3 种大小型号。问这些衬衫有多少种不同的类型？
5. 从纽约到丹佛有 6 条不同的航线，而从丹佛到旧金山有 7 条。如果选一个到丹佛的航班，接着选一个到旧金山的航班，那么从纽约经丹佛到旧金山的旅行有多少种不同的可能性？
6. 从波士顿到底特律有 4 条汽车主干线，而从底特律到洛杉矶有 6 条。那么从波士顿经底特律到洛杉矶的汽车主干线有多少条？
7. 如果用 3 个字母作为姓名的开头，人们可以有多少种不同的选择？
8. 如果这 3 个字母不允许重复，人们可以有多少种不同的选择？
9. 如果这 3 个字母以 A 开始，人们又可以有多少种不同的选择？
10. 8 位二进制串有多少个？
11. 首尾都是 1 的 10 位二进制串有多少个？
12. 位数不超过 6 的二进制串有多少个？
13. 位数不超过 n 的且全由 1 组成的二进制串有多少个？这里的 n 是正整数。
14. 首尾都是 1 的 n 位二进制串有多少个？这里的 n 是正整数。
15. 位数不超过 4 且由小写字母构成的串有多少个？
16. 由 4 个小写字母构成且含有字母 x 的串有多少个？
17. 由 5 个 ASCII 码构成且至少（在符号位）包含一个@字符的串有多少个？（注：有 128 个不同的 ASCII 码。）
18. 有多少个小于 1000 的正整数
    a）被 7 整除？
    b）被 7 整除但不被 11 整除？
    c）同时被 7 和 11 整除？
    d）被 7 或 11 整除？
    e）恰好被 7 或 11 中的一个数整除？
    f）既不被 7 整除，也不被 11 整除？
    g）含有不同的数字？
    h）含有不同的数字且是偶数？
19. 有多少个正整数恰有 3 个十进制数字，即包含在 100 到 999 之间的数
    a）被 7 整除？
    b）是奇数？
    c）有相同的 3 个十进制数字？
    d）不被 4 整除？
    e）被 3 或 4 整除？
    f）不被 3 也不被 4 整除？
    g）被 3 整除但不被 4 整除？
    h）被 3 和 4 整除？

20. 有多少个正整数恰有 4 个十进制数字，即包含在 1000 到 9999 之间的数
   a) 被 9 整除？
   b) 是偶数？
   c) 有不同的十进制数字？
   d) 不被 3 整除？
   e) 被 5 或 7 整除？
   f) 不被 5 也不被 7 整除？
   g) 被 5 整除但不被 7 整除？
   h) 被 5 和 7 整除？
21. 有多少个串含有 3 个十进制数字且
   a) 同一数字不能出现 3 次？
   b) 以奇数字开始？
   c) 恰有 2 个数字是 4？
22. 有多少个串含有 4 个十进制数字且
   a) 同一数字不出现两次？
   b) 以偶数字结束？
   c) 恰有 3 个数字是 9？
23. 一个委员会有 50 个州构成，每个州可从州长或两个参议员中选一个参加，有多少种不同的方式？
24. 用 3 个数字后跟 3 个字母或者 3 个字母后跟 3 个数字可构成多少种车牌？
25. 用 2 个字母后跟 4 个数字或者 2 个数字后跟 4 个字母可构成多少种车牌？
26. 用 3 个字母后跟 3 个数字或者 4 个字母后跟 2 个数字可构成多少种车牌？
27. 用 2 个或 3 个字母后跟 2 个或 3 个数字可构成多少种车牌？
28. 由 8 个英语字母可构成多少个串？
   a) 如果字母可以重复
   b) 如果字母不能重复
   c) 如果字母可以重复且以 X 开始
   d) 如果字母不能重复且以 X 开始
   e) 如果字母可以重复且以 X 开始和结束
   f) 如果字母可以重复且开始于BO(按此次序)
   g) 如果字母可以重复且以BO(按此次序)开始和结束
   h) 如果字母可以重复且以BO(按此次序)开始或结束
29. 由 8 个英语字母可构成多少个串？
   a) 如果字母可以重复且不包含元音字母
   b) 如果字母不能重复且不包含元音字母
   c) 如果字母可以重复且以元音字母开始
   d) 如果字母不能重复且以元音字母开始
   e) 如果字母可以重复且包含至少一个元音字母
   f) 如果字母可以重复且包含恰好一个元音字母

g) 如果字母可以重复且以 X 开始并至少包含一个元音字母

h) 如果字母可以重复且以 X 开始和结束并至少包含一个元音字母

30. 从 10 元素集合到含有下述元素数目的集合有多少个不同的函数?

a) 2 b) 3 c) 4 d) 5

31. 从 5 元素集合到含有下述元素数的集合有多少一对一的函数?

a) 4 b) 5 6) 6 d) 7

32. 从集合{1,2,…,n}到集合{0,1}有多少个函数? 这里的 n 是正整数。

33. 从集合{1,2,…,n}到集合{0,1}有多少个满足下列条件的函数? 这里的 n 是正整数。

a) 是一对一的

b) 对 1 和 n 赋值为 0

c) 对恰好一个小于 n 的正整数赋值为 1

34. 从 5 元素集合到含有下述元素数的集合有多少个部分函数（见 1.6 节的练习)?

a) 1 b) 2 c) 5 d) 9

35. 从 m 元素集合到 n 元素集合有多少个部分函数（见 1.6 节的练习)? 这里的 m 和 n 是正整数。

36. 100 个元素的集合有多少个子集的元素数多于 1?

37. 如果一个字符串反转后所得结果与原来的字符串一样，就称它是一个回文。有多少个长为 n 的二进制串是回文?

38. 在一个婚礼上摄影师从 10 个人中安排 6 个人在一排拍照，其中新娘和新郎也在这 10 个人中，如果满足下述条件，有多少种安排的方式?

a) 新娘必须在照片中

b) 新娘和新郎必须都在照片中

c) 新娘和新郎恰好一个在照片中

39. 在一个婚礼上摄影师安排 6 个人在一排拍照，包含新娘和新郎在内，如果满足下述条件，有多少种安排的方式?

a) 新娘必须在新郎旁边

b) 新娘不在新郎旁边

c) 新娘在新郎左边的某个位置

40. 有多少个 7 位二进制串以 2 个 0 开始或以 3 个 1 结束?

41. 有多少个 10 位二进制串以 3 个 0 开始或以 2 个 0 结束?

*42. 有多少个 10 位二进制串包含 5 个连续的 0 或者 5 个连续的 1?

**43. 有多少个 8 位二进制串包含 3 个连续的 0 或者 4 个连续的 1?

44. 离散数学班的每个学生都是计算机科学或数学专业的，或者是同时修这两个专业的。如果有 38 个是计算机科学专业的（包含同时修两个专业的)，23 个是数学专业的（包含同时修两个专业的)，7 个同时修两个专业的，那么这个班有多少个学生?

45. 有多少个不超过 100 的正整数能被 4 或 6 整除?

46. 在 C 程序设计语言中的变量名是一个字符串，可以包含大写字母、小写字母、数字或下横线。此外，字符串的第一个字符必须是字母（大写或小写字母）或下横线。如果一个变量名由它的前 8 个字符确定，那么在 C 语言中可以命名多少个不同的变量?（注意，

变量名包含的字符数可以少于 8 个)。

47. 假定在将来的某个时间世界上的每部电话将被分配一个号码，这个号码包含一个 1 到 3 位数字的形如 X，XX 或 XXX 的国家代码，后面跟随着一个 10 位数字的形如 $NXX-NXX-XXXX$ 的电话号码（如例 10 所描述的）。在这个编码计划中全世界将有多少个不同的有效电话号码？
48. 使用树图找出不含 3 个连续 0 的 4 位二进制串的个数。
49. 有多少种不同的方式排列字母 a，b，c 和 d，使得 b 不紧跟在 a 的后边？
50. 使用树图找出世界职业棒球大赛可能出现的方式数，其中 7 场中先胜 4 场的队赢得这个比赛。
51. 使用树图确定 $\{3,7,9,11,24\}$ 的子集数使得子集中的元素之和小于 28。

*52. 使用乘积法则证明对于 n 个变量的命题存在 2^{2^n} 个不同的真值表。

53. 从两个任务的求和法则使用数学归纳法证明关于 m 个任务的求和法则。
54. 从两个任务的乘积法则使用数学归纳法证明关于 m 个任务的乘积法则。
55. 具有 n 条边的凸多边形有多少条对角线？（如果在多边形内或边界的每两个顶点的联线完全在这个集合之内，则称为凸多边形。）
56. 因特网中的数据以*数据报*传输，数据报是由二进制位的数据块构成的。每个数据报包含有头信息和数据区。头信息最多被分成 14 个不同的域（详细说明很多事项，包括发送和接受地址），数据区包含被传输的实际数据。14 个头信息域中有一个*头长度域*（表示为 HLEN），根据协议规定是 4 位，它说明了以 32 位为一个数据块的头信息长度。例如，如果 HLEN = 0110，那么头信息由 6 个 32 位的数据块构成。14 个头信息域中的另一个域是 16 位的*总长度域*（表示为 TOTAL LENGTH），它说明了以位为单位的整个数据报包含头信息和数据区在内的总长度。数据区的长度是数据报的总长度减去头的长度。
    a) TOTAL LENGTH 的最大值（16 位长）确定了因特网数据报以字节（8 位的数据块）为单位的最大总长度。这个值是多少？
    b) HLEN 的最大值（4 位长）确定了头信息以 32 位数据块为单位的最大总长度，这个值是多少？以字节为单位的最大的头信息的总长度是多少？
    c) 最小的（最通常的）头长度是 20 字节。因特网数据报的数据区以字节为单位的最大总长度是多少？
    d) 如果头长度是 20 个字节并且总长度是尽可能地长，那么在数据区可以传输多少个不同的字节串？

4.2 鸽巢原理

4.2.1 引言



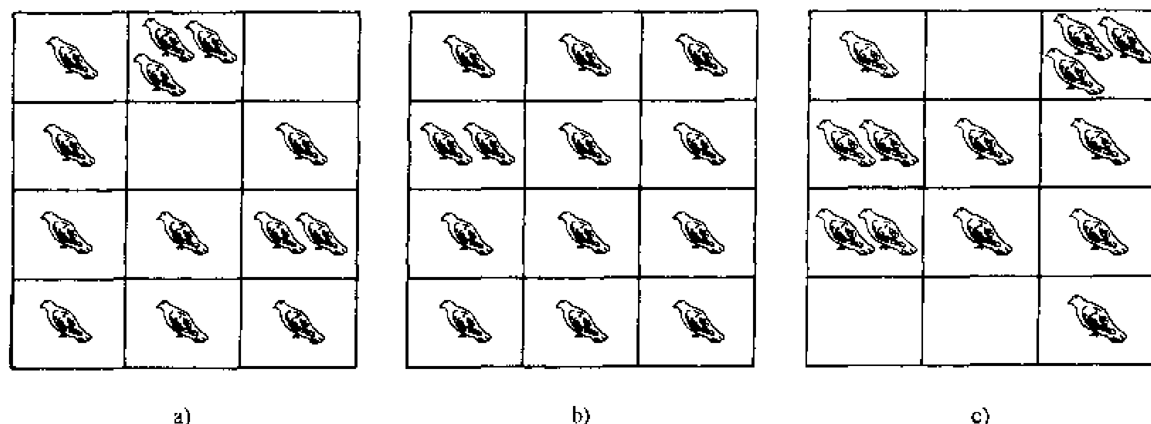
假定一群鸽子飞入一组鸽巢安歇。鸽巢原理表明如果鸽子数比鸽巢数多，那么一定至少有一个鸽巢里至少有 2 只鸽子（见图 4-4)。当然，这个原理除了鸽子和鸽巢外也可以用于其他对象。

定理 1 *鸽巢原理* 如果 $k+1$ 个或更多的物体放入 k 个盒子，那么至少有一个盒子包含了 2 个或更多的物体。

证 假定 k 个盒子中没有一个盒子包含的物体多于 1 个，那么物体总数至多是 k，这与

至少有 $k+1$ 个物体矛盾。 ■

鸽巢原理也叫做狄利克莱抽屉原理，以 19 世纪的德国数学家狄利克莱[1]命名，他经常在工作中使用这个原理。下面的例子说明了鸽巢原理是怎样使用的。



a) b) c)

图 4-4 鸽子比鸽巢多

例 1 在一组 367 个人中一定至少有 2 个人有相同的生日，这是由于只有 366 个可能的生日。 ■

例 2 在 27 个英文单词中一定至少有 2 个单词以同一个字母开始，因为英文字母表中只有 26 个字母。 ■

例 3 如果考试给分是从 0 到 100，班上必须有多少个学生才能保证在这次期终考试中至少有 2 个学生得到相同的分数？

解 期终考试有 101 个分数。鸽巢原理证明在 102 个学生中一定至少有 2 个学生具有相同的分数。 ■

4.2.2 推广的鸽巢原理

鸽巢原理指出当物体比盒子多时一定至少有 2 个物体在同一个盒子里。但是当物体数超过盒子数的倍数时可以叙述更多的结果。例如，在任给 21 个十进制数字中一定有 3 个是相同的。这是由于 21 个物体被分配到 10 个盒子里，那么某个盒子的物体一定多于 2 个。

定理 2 推广的鸽巢原理 如果 N 个物体放入 k 个盒子，那么至少有一个盒子包含了至少 $\lceil N/K \rceil$ 个物体。

证 假定没有盒子包含了比 $\lceil N/k \rceil - 1$ 多的物体，那么物体总数至多是

$$K(\lceil N/k \rceil - 1) < k(((N/k) + 1) - 1) = N$$

这里用到不等式 $\lceil N/k \rceil < (N/k) + 1$。这与存在有总数 N 个物体矛盾。 ■

下面的例子显示了推广的鸽巢原理是怎样使用的。

[1] G. L. 狄利克莱(G. Lejenue Dirichlet, 1805—1859) 狄利克莱诞生在德国科隆附近的一个法国家庭。他在巴黎大学学习并且在布莱斯劳大学和柏林大学工作。1855 年他在哥丁根大学继高斯的职位。据说狄利克莱是第一个掌握 20 年前高斯的“算学研究”的人。据说他手边总有一份抄件，即使旅行时也不例外。狄利克莱在数论中有许多重要的发现，包括在算术级数 $an+b$ 中存在无限多素数的定理，这里的 a 和 b 互素。他对于 $n=5$ 的情况证明了费马的最后的定理，即方程 $x^5+y^5=z^5$ 不存在非平凡的整数解。狄利克莱也对分析作出了许多贡献。

例 4　在 100 个人中至少有$\lceil 100/12 \rceil = 9$ 个人生在同一个月。■

例 5　如果有 5 个可能的成绩 A，B，C，D 和 F，那么在一个离散数学班里最少要多少个学生才能保证至少 6 个学生得到相同的分数？

解　为保证至少 6 个学生得到相同的分数所需的最少学生数是使得$\lceil N/5 \rceil = 6$ 的最小整数 N。这样的最小整数是 $N = 5 \cdot 5 + 1 = 26$。于是，26 是保证至少 6 个学生得到相同的分数所需的最少学生数。■

例 6　为保证一个州的 2500 万个电话有不同的 10 位电话号码所需地区代码的最小数是多少？（假定电话号码是 $NXX - NXX - XXXX$ 形式，其中前 3 位是地区代码，N 表示从 2 到 9 的十进制连续数字，X 表示任何十进制数字）。

解　有 800 万个形如 $NXX - XXXX$ 的不同的电话号码（如 4.1 节的例 10 所显示的）。因此，由推广的鸽巢原理，在 2500 万个电话号码中，至少$\lceil 25\,000\,000 / 8\,000\,000 \rceil$个一定有同样的电话号码。因而至少需要 4 个地区代码来保证所有的 10 位号码是不同的。■

4.2.3　巧妙使用鸽巢原理

在鸽巢原理的许多有趣应用中必须用某种巧妙的方式选择放入盒子的物体。下面将描述这样的一些应用。

例 7　在 30 天的一个月里，某棒球队一天至少打一场比赛，但至多打 45 场。证明一定有连续的若干天内这个队恰好打了 14 场。

解　令 a_j 是在这个月的第 j 天或第 j 天之前所打的场数，则 a_1，a_2，…，a_{30}是不同正整数的一个递增序列，其中 $1 \leqslant a_j \leqslant 45$。从而 $a_1 + 14$，$a_2 + 14$，…，$a_{30} + 14$ 也是不同正整数的一个递增序列，其中 $15 \leqslant a_j + 14 \leqslant 59$。

60 个正整数 a_1，a_2，…，a_{30}，$a_1 + 14$，$a_2 + 14$，…，$a_{30} + 14$ 全都小于或等于 59。因此，由鸽巢原理有两个正整数相等。因为整数$a_j (j = 1, 2, \cdots, 30)$都不相同，并且$a_j + 14 (j = 1, 2, \cdots, 30)$也不相同，一定存在下标 i 和 j 满足 $a_i = a_j + 14$。这意味着从第 $j+1$ 天到第 i 天恰好打了 14 场比赛。■

例 8　证明在不超过 $2n$ 的任意 $n+1$ 个正整数中一定存在一个正整数被另一个正整数整除。

解　把 $n+1$ 个整数 a_1，a_2，…，a_n 中的每一个都写成 2 的幂与一个奇数的乘积。换句话说，令 $a_j = 2^{k_j} q_j$，$j = 1, 2, \cdots, n+1$，其中 k_j 是非负整数，q_j 是奇数。整数 q_1，q_2，…，q_{n+1}都是小于 $2n$ 的正奇数。因为只存在 n 个小于 $2n$ 的正奇数，由鸽巢原理，q_1，q_2，…，q_{n+1}中必有两个相等。于是，存在整数 i 和 j 使得 $q_i = q_j$。令 q_i 与 q_j 的公共值是 q，那么 $a_i = 2^{k_i} q$，$a_j = 2^{k_j} q$。因而，若 $k_i < k_j$，则 a_i 整除 a_j；若 $k_i > k_j$，则 a_j 整除 a_i。■

巧妙的应用鸽巢原理证明了在不同整数的序列中存在着确定长度的递增或递减子序列。在给出这个应用之前先回顾某些定义。假定 a_1，a_2，…，a_N 是实数序列。它的一个子序列是形为 a_{i_1}，a_{i_2}，…，a_{i_m} 的序列，其中 $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant N$。因此一个子序列是从初始

序列得到的序列，按照原来的顺序选取初始序列的某些项，也许要排除其他的项。如果这个序列的每一项都大于它前面的项，就称为*严格递增的*；如果每一项都小于它前面的项，就称为*严格递减的*。

定理3 每个由 n^2+1 个不同实数构成的序列都包含一个长为 $n+1$ 的严格递增子序列或严格递减子序列。

在定理证明之前先给出下面的例子。

例9 序列8，11，9，1，4，6，12，10，5，7包含10项。注意到 $10=3^2+1$，存在四个长为4的递增子序列，即1，4，6，12；1，4，6，7；1，4，6，10和1，4，5，7。也存在一个长为4的递减子序列，即11，9，6，5。 ■

现在给出定理的证明。

证 令 $a_1, a_2, \cdots, a_{n^2+1}$ 是 n^2+1 个不同实数的序列。和序列中的每一项 a_k 联系着一个有序对，即 (i_k, d_k)，其中 i_k 是从 a_k 开始的最长的递增子序列的长度，且 d_k 是从 a_k 开始的最长的递减子序列的长度。

假定没有长为 $n+1$ 的递增或递减子序列。那么 i_k 和 d_k 都是小于或等于 n 的正整数，$k=1, 2, \cdots, n^2+1$。因此，由乘积法则，关于 (i_k, d_k) 存在 n^2 个可能的有序对。根据鸽巢原理，n^2+1 个有序对中必有两个相等。换句话说，存在项 a_s 和 a_t，$s<t$，使得 $i_s=i_t$ 和 $d_s=d_t$。我们将证明这是不可能的。由于序列的项是不同的，不是 $a_s<a_t$ 就是 $a_s>a_t$。如果 $a_s<a_t$，那么由于 $i_s=i_t$，那么把 a_s 加到从 a_t 开始的递增子序列前面就构造出一个从 a_s 开始的长为 i_t+1 的递增子序列。从而产生矛盾。类似地，如果 $a_s>a_t$，可以证明 d_s 一定大于 d_t，从而也产生矛盾。 ■

最后的例子说明了怎样把推广的鸽巢原理用于组合数学的重要部分拉姆赛理论，它是以英国数学家拉姆赛[○]而命名的。一般地说，拉姆赛理论可用于处理集合元素的子集分配问题。

例10 假定一组有6个人，任意两个人或者是朋友或者是敌人。证明在这组人中或存在3个人彼此都是朋友，或存在3个人彼此都是敌人。

解 令 A 是6个人中的一人，组里其他5个人中至少有3个人是 A 的朋友，或至少有3个人是 A 的敌人。这可从推广的鸽巢原理得出，因为当5个物体被分成两个集合时，其中的一个集合至少有 $\lceil 5/2 \rceil=3$ 个元素。若是前一种情况，假定 B，C 和 D 是 A 的朋友。如果这3个人中有2个人也是朋友，那么这2个人和 A 构成彼此是朋友的3人组。否则，B，C 和 D 构成彼此为敌人的3人组。对于后一种情况的证明，当 A 存在3个或更多的敌人时可以用类似的方法处理。 ■

○ 富兰克·波拉姆顿·拉姆赛（Frank Plumpton Ramsey，1903—1930）拉姆赛是剑桥马格达林学院校长的儿子，在温彻斯特和特里尼特学院受过教育。1923年毕业以后，他应聘在剑桥皇家学院工作，并在那里度过余生。拉姆赛对数理逻辑作出了重要的贡献。我们现在所称的拉姆赛理论是由他在“一个形式逻辑问题”的论文中所发表的聪明的组合论辩引起的。拉姆赛也对经济数学理论作出了贡献。他作为在数学基础方面的优秀讲师而受到注意。他死于26岁，他的死使得数学界和剑桥大学失去了一个才华横溢的年轻学者。

练习

1. 假定周末不排课，证明在任一组 6 门课中一定有 2 门课安排在同一天上课。
2. 如果一个班有 30 个学生，证明至少 2 个学生的姓以同一个字母开头。
3. 抽屉里有一打棕色的短袜和一打黑色的短袜，全都没有配好对。一个人在黑暗中随机取出一些袜子。
   a）必须取多少只袜子才能保证至少有 2 只袜子是同色的？
   b）必须取多少只袜子才能保证至少有 2 只袜子是黑色的？
4. 一个碗里有 10 个红球和 10 个蓝球。一个女士不看着球而随机地选取。
   a）她必须选多少个球才能保证至少有 3 个球是同色的？
   b）她必须选多少个球才能保证至少有 3 个球是蓝色的？
5. 证明在任意 5 个整数中（不一定是连续的）有 2 个整数被 4 除的余数相等。
6. 设 d 是正整数，证明在任意一组 $d+1$ 个整数中（不一定是连续的）有 2 个整数被 d 除的余数相等。
7. 设 n 是正整数。证明在任意一组 n 个连续的正整数中恰好有 1 个被 n 整除。
8. 证明如果 f 是从 S 到 T 的函数，其中 S 和 T 是有穷集，满足 $|S|>|T|$，那么在 S 中存在元素 s_1 和 s_2 使得 $f(s_1)=f(s_2)$，或者换句话说，f 不是一对一的。
9. 在一个大学里每个学生来自 50 个州中的一个州，那么必须有多少个学生注册才能保证至少有 100 个学生来自同一个州？

*10. 设 $(x_i, y_i)(i=1,2,3,4,5)$ 是 xy 平面上一组具有整数坐标的 5 个不同的点。证明至少有一对点的联线中点的坐标是整数。

*11. 设 $(x_i, y_i, z_i)(i=1,2,3,4,5,6,7,8,9)$ 是 xyz 空间中一组具有整数坐标的 9 个不同的点。证明至少有一对点的联线中点的坐标是整数。

12. 至少需要多少个有序对 (a,b) 才能保证存在两个有序对 (a_1,b_1) 和 (a_2,b_2)，使得 $a_1 \bmod 5 = a_2 \bmod 5$，并且 $b_1 \bmod 5 = b_2 \bmod 5$。
13. a）如果从前 8 个正整数中选 5 个整数一定存在一对整数其和等于 9。
    b）如果不是选 5 个而是选 4 个整数，a)的结论还正确吗？
14. a）如果从前 10 个正整数中选 7 个整数一定至少存在 2 对整数其和等于 11。
    b）如果不是选 7 个而是选 6 个整数，a)的结论还正确吗？
15. 一个公司在仓库储存产品。仓库中的存储柜由它们的通道、在通道中的位置和货架来指定。整个仓库有 50 个通道，每个通道 85 个水平位置，每个位置 5 个货架。公司产品数至少是多少才能使得在同一个存储柜中至少有 2 个产品？
16. 一条街道上有 51 所房子，每所房子的地址在 1000 到 1099 之间（1000 与 1099 包括在内）。证明至少有 2 所房子的地址是连续的。

*17. 设 x 是无理数。证明对于某个不超过 n 的正整数 j，在 jx 和到 jx 最近的整数之间的差的绝对值小于 $1/n$。

18. 在序列 22，5，7，2，23，10，15，21，3，17 中找出一个最长的递增子序列和一个最长的递减子序列。
19. 构造 16 个正整数的序列使得它没有 5 项的递增或递减子序列。

20. 如果 101 个不同身高的人站在一条线上，证明可能找到 11 个人，他们站在线上的高度是按递增或者递减顺序。

*21. 用伪码描述一个算法产生一个不同整数的序列的最大递增或递减子序列。

22. 证明在任一组 5 个人中（其中任两个人或者是朋友或者是敌人），不一定有 3 个人彼此都是朋友或者 3 个人彼此都是敌人。

23. 证明在任一组 10 个人中（其中任两个人或者是朋友或者是敌人），或存在 3 个人彼此都是朋友，或存在 4 个人彼此都是敌人，并且存在 3 个人彼此是敌人，或存在 4 个人彼此是朋友。

24. 使用练习 23 证明在任一组 20 个人中（其中任两个人或者是朋友或者是敌人），或存在 4 个人彼此都是朋友，或存在 4 个人彼此都是敌人。

25. 证明在加里福尼亚州（人口 2500 万）至少有 4 个人姓的前 3 个字母相同，并且他们生在一年的同一天（但不一定是同一年）。

26. 证明如果美国工薪阶层有 100 000 000 人的工资低于 1 000 000 美元，那么去年有 2 个人的工资恰好相同（准确到美分）。

27. 一个大学有 38 个不同的时间段来安排课程，如果有 677 门不同的课程，那么需要多少个不同的教室？

28. 一个计算机网络由 6 台计算机组成。每台计算机至少直接连接到一台其他的计算机。证明网络中至少有两台计算机直接连接相同数目的其他计算机。

29. 一个计算机网络由 6 台计算机组成。每台计算机直接连接到零台或者更多的其他计算机。证明网络中至少有两台计算机直接连接相同数目的其他计算机。

*30. 证明在至少 2 个人的聚会中，存在 2 个人认识人数相同的其他人。

31. 一个摔跤选手是 75 小时之内的冠军。该选手一小时至少赛一场，但总共不超过 125 场。证明存在着连续的若干个小时使得该选手恰好进行了 24 场比赛。

*32. 如果练习 31 中的 24 被替换如下，命题是否为真？

a) 2　　b) 23　　c) 25　　d) 30

33. 如果 f 是从 S 到 T 的函数，其中 S 和 T 是有穷集，并且 $m=\lceil |S|/|T| \rceil$，那么证明至少存在 S 的 m 个元素映到 T 的同一个值。即存在 S 中的元素 s_1，s_2，…，s_m 使得 $f(s_1)=f(s_2)=\cdots=f(s_m)$。

34. 假定一个小学院的离散数学班中有 9 个学生。

a) 证明这个班一定至少有 5 个男学生或至少有 5 个女学生。

b) 证明这个班一定至少有 3 个男学生或至少有 7 个女学生。

35. 假定在 25 个学生的离散数学班中每个学生是一年级、二年级或三年级学生。

a) 证明在这个班里至少有 9 个一年级学生，至少 9 个二年级学生，或至少 9 个三年级学生。

b) 证明在这个班里至少有 3 个一年级学生，至少 19 个二年级学生，或至少 5 个三年级学生。

36. 设 n_1，n_2，…，n_t 是正整数。证明如果 $n_1+n_2+\cdots+n_t-t+1$ 个物体放到 t 个盒子里，则对某个 $i(i=1,2,\cdots,t,)$ 第 i 个盒子包含了至少 n_i 个物体。

*37. 以推广的鸽巢原理为基础的定理 3 的证明概述在这个练习中。使用的记号与教科书中

的证明一样。

a) 假定 $i_k \leqslant n$，$k=1, 2, \cdots, n^2+1$。使用推广的鸽巢原理证明存在 $n+1$ 个项 a_{k_1}，a_{k_2}，$\cdots$，$a_{k_{n+1}}$满足 $i_{k_1}=i_{k_2}=\cdots=i_{k_{n+1}}$，其中 $1\leqslant k_1<k_2<\cdots<k_{n+1}$。

b) 证明 $a_{k_j}>a_{k_{j+1}}$，$j=1, 2, \cdots, n$。[提示：假定 $a_{k_j}<a_{k_{j+1}}$，证明这将推出 $i_{k_j}>i_{k_{j+1}}$的矛盾。]

c) 使用 a) 和 b) 证明，如果没有长为 $n+1$ 的递增子序列，那么一定有同样长的递减子序列。

4.3 排列与组合

4.3.1 引言

假定一个网球队有 10 个选手，教练必须选择 5 个选手外出到另一个学校参加比赛。此外，教练还必须准备 4 个选手的排名表来进行 4 场单打比赛。在这一节，将提出一些方法来计数无序挑选 5 名外出比赛选手的各种组合和参加 4 场单打比赛的不同的 4 个选手的排名表。更一般地，将引入计数一个有穷集的不同个体的无序选择和有序安排的技术。

4.3.2 排列



不同个体集合的一个排列是这些个体的一种有序安排。我们也对集合的某些元素的有序安排感兴趣。一个集合的 r 个元素的有序安排叫做 r-排列。

例 1 设 $S=\{1,2,3\}$。3，1，2 是 S 的一个排列，3，2 是 S 的一个 2-排列。 ■

一个 n 元集的 r-排列数记为$P(n,r)$。我们可以使用乘积法则求出$P(n,r)$。

定理 1 具有 n 个不同元素的集合的 r-排列数是

$$P(n,r)=n(n-1)(n-2)\cdots(n-r+1)$$

证 选择这个排列的第一个元素可以有 n 种方法，因为集合中有 n 个元素。选择排列的第二个元素有 $n-1$ 种方法，由于在使用了为第一个位置挑出的元素之后集合里还留下了 $n-1$个元素。类似地，选择第三个元素有 $n-2$ 种方法，依此类推，直到选择第 r 个元素恰好有 $n-r+1$ 种方法。因此，由乘积法则，存在

$$n(n-1)(n-2)\cdots(n-r+1)$$

个集合的 r-排列。 ■

从定理 1 得出

$$P(n,r)=n(n-1)(n-2)\cdots(n-r+1)=n!/(n-r)!$$

特别的有$P(n,n)=n!$。我们将用一些例子说明这个结果。

例 2 有多少种不同的方式从某个网球队的 10 个选手中挑选 4 个不同的选手参加 4 场网球比赛？注意这里说的比赛是被排序的。

解 答案是 10 元集的 4-排列数。由定理 1，结果是 $P(10, 4)=10\cdot9\cdot8\cdot7=5040$。 ■

例 3 假定有 8 个赛跑运动员。第一名得到一枚金牌，第二名得到一枚银牌，第三名得到一枚铜牌。如果比赛可能出现所有可能的结果，有多少种不同的颁奖方式？

解 颁奖方式就是 8 元集的 3-排列数。因此存在 $P(8,3)=8\cdot7\cdot6=336$ 种可能的颁奖方式。■

例 4 假定一个女推销员要访问 8 个不同的城市。她的访问必须从某个指定的城市开始，但对其他 7 个城市的访问可以按照她想要的任何次序进行。当访问这些城市时，这个女推销员可以有多少种可能的次序？

解 由于第一个城市是确定的，而其他 7 个城市可以是任意的顺序，故城市之间可能的路径数是 7 个元素的排列数。因此，这个女推销员有 $7!=7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1=5040$ 种方式选择她的旅行。比如说，如果这个女推销员想要在城市中找出具有最短距离的路径，并且她对每一条可能的路径计算总距离，那么她必须考虑 5040 条路径。■

4.3.3 组合

集合元素的一个 r-组合是从这个集合无序选取的 r 个元素。于是，简单地说，一个 r-组合是这个集合的一个 r 个元素的子集。

例 5 设 S 是集合 $\{1,2,3,4\}$，那么 $\{1,3,4\}$ 是 S 的一个 3-组合。■

具有 n 个不同元素集合的 r-组合数记为 $C(n,r)$。

例 6 因为 $\{a,b,c,d\}$ 的 2-组合是 $\{a,b\}$ $\{a,c\}$ $\{a,d\}$ $\{b,c\}$ $\{b,d\}$ 和 $\{c,d\}$ 6 个子集，我们有 $C(4,2)=6$。■

我们可以用关于集合的 r-排列数的公式确定 n 元集的 r-组合数。为此只需注意到集合的 r-排列可以按下述方法得到：首先构成集合的 r-组合，接着排列这些组合中的元素。下面的定理给出了 $C(n,r)$ 的值，它的证明就是基于这个观察。

定理 2 设 n 是正整数，r 是满足 $0\leqslant r\leqslant n$ 的整数，n 元素集合的 r-组合数等于

$$C(n,r)=\frac{n!}{r!\,(n-r)!}$$

证 可以如下得到这个集合的 r-排列。先构成集合的 $C(n,r)$ 个 r-组合，然后以 $P(n,r)$ 种方式排序每个 r-组合中的元素，这可以用 $P(r,r)$ 种方式来做。因此，

$$P(n,r)=C(n,r)\cdot P(r,r)$$

这就推出

$$C(n,r)=\frac{P(n,r)}{P(r,r)}=\frac{n!/(n-r)!}{r!/(r-r)!}=\frac{n!}{r!(n-r)!}$$ ■

下面的推论对计算一个集合的 r-组合数是有帮助的。

推论 1 设 n 和 r 是满足 $r\leqslant n$ 的非负整数，那么 $C(n,r)=C(n,n-r)$。

证 由定理 2 得到

$$C(n,r)=\frac{n!}{r!(n-r)!}$$

和

$$C(n,n-r)=\frac{n!}{(n-r)![n-(n-r)!]}=\frac{n!}{(n-r)!r!}$$

因此，$C(n,r)=C(n,n-r)$。 □

对一个 n 元素集合的 r-组合数也有另一种常用的记号，即

$$\binom{n}{r}$$

这个数也叫做*二项式系数*。使用二项式系数这个名字是由于这些数作为系数出现在形如 $(a+b)^n$ 的二项式幂的展开式中。在这一节的后面我们将讨论二项式定理，它把一个二项式的幂表示成与二项式系数有关的项之和。

例 7 有多少种方式从 10 个选手的网球队中选择 5 个选手外出参加在另一个学校的比赛?

解 答案由 10 元素集合的 5-组合数给出。根据定理 2，这个组合数是

$$C(10,5)=\frac{10!}{5!5!}=252$$ ■

例 8 为发展学校的离散数学课程要选出一个委员会。如果数学系有 9 个教师，计算机科学系有 11 个教师。而这个委员会要由 3 个数学系的教师和 4 个计算机科学系的教师组成，那么有多少种选择的方式?

解 由乘积法则，答案是 9 元素集合的 3-组合数与 11 元素集合的 4-组合数之积。根据定理 2，选择这个委员会的方式数是

$$C(9,3)\cdot C(11,4)=\frac{9!}{3!6!}\cdot\frac{11!}{4!7!}=84\cdot 330=27720$$ ■

4.3.4 二项式系数

本节将讨论二项式系数的某些更重要的性质。要讨论的第一个性质是一个重要的恒等式。

定理 3 *帕斯卡恒等式* 设 n 和 k 是满足 $n\geqslant k$ 的正整数，那么有

$$C(n+1,k)=C(n,k-1)+C(n,k)$$

证 假定 T 是包含 $n+1$ 个元素的集合。令 a 是 T 的一个元素且 $S=T-\{a\}$。注意到 T 的包含 k 个元素的子集有 $C(n+1,k)$ 个。然而 T 的含 k 个元素的子集或者包含 a 和 s 中的 $k-1$ 个元素，或者不包含 a 但包含 S 中的 k 个元素。由于 S 的 $k-1$ 元素子集有 $C(n,k-1)$ 个，故 T 含 a 在内的 k 元素子集有 $C(n,k-1)$ 个。又由于 S 的 k 元素子集有 $C(n,k)$ 个，故 T 的不含 a 的 k 元子集有 $C(n,k)$ 个。从面得到，

$$C(n+1,k)=C(n,k-1)+C(n,k)$$ □

注意 *这里给出了帕斯卡恒等式的一个组合证明。也可以从关于 $C(n,r)$ 的公式通过代数推导来证明这个恒等式（见节末的练习 47）。*

帕斯卡恒等式是二项式系数以三角形表示的几何排列的基础，如图 4-5 所示。

这个三角形的第 n 行由二项式系数

$$\binom{n}{k}, k=0,1,\cdots,n$$

组成。

这个三角形叫做帕斯卡[⊖]三角形。帕斯卡恒等式证明，当这个三角形中两个相邻的二项式系数相加时，就产生了下一行在这些两个系数之间的二项式系数。

除了帕斯卡恒等式以外，二项式系数也有许多其他的恒等式。这里只叙述另外两个恒等式，同时给出组合证明。其余的恒等式可以在节后的练习中找到。

定理4　设 n 是正整数，那么

$$\sum_{k=0}^{n} C(n,k)=2^n$$

证　n 元素集合总共有 2^n 个不同的子集。每个子集中有 0 个，1 个，2 个，…或 n 个元素。0 元素子集有 $C(n,0)$ 个，1 元素子集有 $C(n,1)$ 个，2 元素子集有 $C(n,2)$ 个，…，而 n 元素子集有 $C(n,n)$ 个。因此，

$$\sum_{k=0}^{n} C(n,k)$$

$$\binom{0}{0}$$
$$\binom{1}{0}\ \binom{1}{1}$$
$$\binom{2}{0}\ \binom{2}{1}\ \binom{2}{2}$$
$$\binom{3}{0}\ \binom{3}{1}\ \binom{3}{2}\ \binom{3}{3}$$
$$\binom{4}{0}\ \binom{4}{1}\ \binom{4}{2}\ \binom{4}{3}\ \binom{4}{4}$$
$$\binom{5}{0}\ \binom{5}{1}\ \binom{5}{2}\ \binom{5}{3}\ \binom{5}{4}\ \binom{5}{5}$$
$$\binom{6}{0}\ \binom{6}{1}\ \binom{6}{2}\ \binom{6}{3}\ \binom{6}{4}\ \binom{6}{5}\ \binom{6}{6}$$
$$\binom{7}{0}\ \binom{7}{1}\ \binom{7}{2}\ \binom{7}{3}\ \binom{7}{4}\ \binom{7}{5}\ \binom{7}{6}\ \binom{7}{7}$$
$$\binom{8}{0}\ \binom{8}{1}\ \binom{8}{2}\ \binom{8}{3}\ \binom{8}{4}\ \binom{8}{5}\ \binom{8}{6}\ \binom{8}{7}\ \binom{8}{8}$$
…

a)

由帕斯卡恒等式

$$\binom{6}{4}+\binom{6}{5}=\binom{7}{5}$$

1
1　1
1　2　1
1　3　3　1
1　4　6　4　1
1　5　10　10　5　1
1　6　15　20　15　6　1
1　7　21　35　35　21　7　1
1　8　28　56　70　56　28　8　1
…

b)

图 4-5　帕斯卡三角形

⊖ 帕斯卡（*Blaise Pascal*，1623—1662）　帕斯卡在幼年时就显现出他的才能，虽然他的父亲（对解析几何有过多项建树）为了鼓励他在其他方面的兴趣，不让他接触数学书。帕斯卡 16 岁时发现了圆锥曲线中的一个重要结果。18 岁他设计了一部计算机，建造后将其出卖。帕斯卡和费尔马一道奠定了现代概率论的基础。在他的工作中有对现今称为帕斯卡三角形的一些发现。1654 年帕斯卡放弃了对数学的追求，转而研究神学。在那以后，他只有一次重返数学。一个晚上，因剧烈牙痛而心烦意乱，他想通过研究摆线性质来缓解疼痛。不可思议的是牙痛居然减退了，他把这一点看成上天赞成研究数学的暗示。

计数了 n 元素集合的子集的总数，从而证明了

$$\sum_{k=0}^{n} C(n,k) = 2^n$$

定理 5 范德蒙德恒等式 设 m，n 和 r 是非负整数，其中 r 不超过 m 或 n，那么

$$C(m+n,r) = \sum_{k=0}^{r} C(m,r-k)C(n,k)$$

注意 这个恒等式是由 18 世纪的数学家亚历山大－舍费尔·范德蒙德[⊖]发现的。

证 假定在第一集合中有 m 个项，第二集合中有 n 个项。从这两个集合的并集中取 r 个元素的方式数是 $C(m+n,r)$。从并集中取 r 个元素的另一种方式是先从第一个集合中取 k 个元素，接着从第二个集合中取 $r-k$ 个元素，其中 k 是满足 $0 \leqslant k \leqslant r$ 的整数。用乘积法则，这可以用 $C(m,k)C(n,r-k)$ 种方式完成。所以，从这个并集中选取 r 个元素的总方式数等于

$$C(m+n,r) = \sum_{k=0}^{r} C(m,r-k)C(n,k)$$

这就证明了范德蒙德恒等式。

4.3.5 二项式定理

二项式定理给出了二项式幂的展开式的系数。一个二项式只不过是两项的和，例如 $x+y$。(这些项可以是常数与变量的积，但这里先不考虑。)下面的例子说明为什么有这个定理。

例 9 $(x+y)^3$ 的展开式可以使用组合推理而不是用三个项的乘法得到。当 $(x+y)^3 = (x+y)(x+y)(x+y)$ 被展开时，把所有由第一个和的一项、第二个和的一项与第三个和的一项产生的乘积加起来。从而出现了形如 x^3，x^2y，xy^2 和 y^3 的项。为得到形如 x_3 的项，在每个和里必须选择一个 x，只有一种方式能做到这一点。因此，乘积中 x^3 项的系数是 1。为得到形如 x^2y 的项，必须从三个和中的两个和里选 x（而因此在另一个和里选 y）。于是，这种项的个数是三个个体的 2-组合数，即 $C(3,2)$。类似地，形如 xy^2 项的个数是三个和中选一个来提供 x 的方式数（而另两个和中都要选 y），有 $C(3,1)$ 种方式能够做到这一点。最后，得到 y^3 的唯一的方式是三个和的每一个都选择 y，恰好有一种方式能够做到这一点。因此得到

$$(x+y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$$ ■

现在叙述二项式定理。

定理 6 二项式定理 设 x 和 y 是变量，n 是正整数，那么

⊖ 亚历山大－舍费尔·范德蒙德（Alexandre－Théophile Vandermonde，1735—1796）范德蒙德由于是一个体弱多病的孩子，他的医生父亲让他从事音乐职业。但是后来，他对数学越来越感兴趣。他的完整的数学工作包含在 1771—1772 年发表的 4 篇论文中。这些论文包括了在方程求根、行列式理论以及骑士旅行问题（在 7.5 节的练习中介绍）的基础贡献。范德蒙德对数学的兴趣只持续了两年。后来，他在和声学、寒冷实验以及钢的制造等方面发表论文。他也对政治发生了兴趣，参加法国革命事业，并且在政府中担任了几个不同的职务。

$$
\begin{aligned}
(x+y)^n &= \sum_{j=0}^{n} C(n,j)x^{n-j}y^j \\
&= \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \binom{n}{2}x^{n-2}y^2 + \cdots + \binom{n}{n-1}xy^{n-1} + \binom{n}{n}y^n
\end{aligned}
$$

证 这里给出定理的组合证明。当乘积被展开时其中的项都是下述形式：$x^{n-j}y^j$，$j=0, 1, 2, \cdots, n$。为计数形如 $x^{n-j}y^j$ 的项数，观察到必须从 n 个和中选 $n-j$ 个 x（从而乘积中其他的 j 个项都是 y）才能得到这种项。因此，$x^{n-j}y^j$ 的系数是 $C(n,n-j)=C(n,j)$。定理得证。

下面的例子给出二项式定理的应用。

例 10 $(x+y)^4$ 的展开式是什么？

解 由二项式定理得到

$$
\begin{aligned}
(1+x)^4 &= \sum_{j=0}^{4} C(4,j)x^{4-j}y^j \\
&= C(4,0)x^4 + C(4,1)x^3y + C(4,2)x^2y^2 + C(4,3)xy^3 + C(4,4)y^4 \\
&= x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4
\end{aligned}
$$
■

例 11 在 $(x+y)^{25}$的展开式中 $x^{12}x^{13}$的系数是什么？

解 由二项式定理得到这个系数是

$$C(25,13) = \frac{25!}{13!\ 12!} = 5\ 200\ 300$$
■

例 12 在 $(2x-3y)^{25}$的展开式中 $x^{12}y^{13}$的系数是什么？

解 首先注意到这个表达式等于$(2x+(-3y))^{25}$。由二项式定理，我们有

$$(2x+(-3y))^{25} = \sum_{j=0}^{25} C(25,j)(2x)^{25-j}(-3y)^j$$

因此，当 $j=13$ 时得到展开式中 $x^{12}y^{13}$的系数，即

$$C(25,13)2^{12}(-3)^{13} = -\frac{25!}{13!\ 12!}2^{12}3^{13}$$
■

可用二项式定理给出定理 4 的另一个证明。这个定理指出，只要 n 是正整数，就有

$$\textstyle\sum_{k=0}^{n} C(n,k) = 2^n$$

证 使用二项式定理可以看出

$$2^n = (1+1)^n = \textstyle\sum_{k=0}^{n} C(n,k)1^k1^{n-k} = \sum_{k=0}^{n} C(n,k)$$

这正是需要的结果。

二项式定理也可以用于证明下面的恒等式。

定理7 设 n 是正整数，那么

$$\sum_{k=0}^{n} (-1)^k C(n,k) = 0$$

证 由二项式定理得出

$$0=((-1)+1)^n=\sum_{k=0}^{n}C(n,k)(-1)^k1^{n-k}=\sum_{k=0}^{n}C(n,k)(-1)^k$$ □

从而证明了定理。

练习

1. 列出$\{a,b,c\}$的所有排列。
2. 集合$\{a,b,c,d,e,f,g\}$有多少个排列？
3. $\{a,b,c,d,e,f,g\}$有多少个排列以 a 结尾？
4. 令$S=\{1,2,3,4,5\}$。
   a) 列出 S 的所有 3-排列。
   b) 列出 S 的所有 3-组合。
5. 求出下面每个排列数的值
   a) $P(6,3)$　b) $P(6,5)$　c) $P(8,1)$
   d) $P(8,5)$　e) $P(8,8)$　f) $P(10,9)$
6. 求出下面每个组合数的值
   a) $C(5,1)$　b) $C(5,3)$　c) $C(8,4)$
   d) $C(8,8)$　e) $C(8,0)$　f) $C(12,6)$
7. 求出 9 元素集合的 5-排列数。
8. 如果不允许并列名次，在结束比赛时 5 个赛跑运动员有多少种不同的排名次序？
9. 在一场 12 匹马的赛马中，如果所有的比赛结果都是可能的，对于第一名、第二名和第三名有多少种可能性？
10. 有 6 个不同的人竞选州长。有多少种不同的次序在选票上打印竞选者的名字？
11. 一个组有 n 个男士和 n 个女士。如果把他们男女相间地排成一排，有多少种方式？
12. 有多少种不同的方式选择两个小于 100 的正整数？
13. 有多少种不同的方式从英语字母表中选择 5 个字母？
14. 一个 10 元素集合有多少个子集含有奇数个元素？
15. 一个 100 个元素的集合有多少个子集包含的元素多于 2 个？
16. 有多少个 10 位二进制串
    a) 恰好有 3 个 0？
    b) 0 和 1 的数目相等？
    c) 至少有 7 个 1？
    d) 至少有 3 个 1？
17. 把编号为 1，2，…，100 的 100 张票卖给 100 个不同的人来抽奖。有 4 项不同的奖，包括 1 项大奖（到塔希提岛旅游）。如果满足下面的条件，有多少种不同的抽奖方式？
    a) 没有限制。
    b) 拿 47 号票的人赢了大奖。
    c) 拿 47 号票的人赢了一项奖。
    d) 拿 47 号票的人没赢奖。
    e) 拿 19 和 47 号票的人都赢了奖。

f) 拿 19，47 和 73 号票的人都赢了奖。

g) 拿 19，47，73 和 97 号票的人都赢了奖。

h) 拿 19，47，73 和 97 号票的人都没赢奖。

i) 拿 19，47，73 或 97 号票的人赢了大奖。

j) 拿 19 和 47 号票的人赢了奖，但拿 73 和 97 号票的人没有赢奖。

18. 一个垒球队的 13 个人参加一场比赛。

a) 有多少种方式选 10 个选手上场？

b) 有多少种方式从 13 个参赛的人中分配 10 个选手的位置？

c) 13 个出席的人中有 3 个女士。如果上场的选手中必须要求至少有一个女士，那么有多少种方式选择 10 个选手？

19. 一个俱乐部有 25 个成员。

a) 有多少种方式从中选择 4 个人作为董事会成员。

b) 有多少种方式从中选出俱乐部的主席、副主席、书记和司库？

20. 一个教授写了 40 道离散数学的真假判定题。在这些题中有 17 个命题为真。如果可以按照任意次序排列这些题，可能有多少种不同的答案？

21. 用不超过 100 的正整数构成 4-排列，其中有多少个排列包含 3 个连续的整数？

a) 这里的连续指按照整数通常的顺序，并且这些连续整数可能被排列中的其他整数分开。

b) 这里的连续不但指整数是连续的，而且它们在排列中的位置也是连续的。

22. 一所学校的数学系有 7 名女教师和 9 名男教师。

a) 有多少种方式从中选出 5 人的委员会，并使其中包含至少 1 名女教师？

b) 有多少种方式从中选出 5 人的委员会，并使其中包含至少 1 名女教师和至少一名男教师。

23. 英语字母表中包含 21 个辅音和 5 个元音。由英语字母表的 6 个小写字母可构成多少字符串使得它们包含

a) 恰好 1 个元音？

b) 恰好 2 个元音？

c) 至少 1 个元音？

d) 至少 2 个元音？

24. 由英语字母表中的 6 个小写字母可构成多少字符串使得它们包含

a) 字母 a？

b) 字母 a 和 b？

c) 字母 a 和 b，其中 a 在 b 前边的邻接位置，同时所有的字母都不相同？

d) 字母 a 和 b，其中 a 在 b 左边的某个位置，同时所有的字母都不相同？

25. 假定某个系包含 10 名男士和 15 名女士。有多少种方式组成一个 6 人委员会且使得它含有相同数量的男士和女士？

26. 假定某个系包含 10 名男士和 15 名女士。有多少种方式组成一个 6 人委员会且使得它含有的女士比男士多？

27. 有多少个二进制串恰好包含 8 个 0 和 10 个 1，如果每个 0 后面必须紧跟着一个 1？

28. 有多少个二进制串恰好包含 5 个 0 和 14 个 1，如果每个 0 后面必须紧跟着两个 1？

29. 有多少个 10 位二进制串包含至少 3 个 1 和至少 3 个 0?
30. 有多少种方式从联合国中选择 12 个国家成为理事国且使得 3 个选自 45 个国家的一组，4 个选自 57 个国家的一组，其他的选自剩下的 69 个国家?
31. 有多少种方式用 3 个字母后跟 3 个数字组成汽车牌照并且没有一个字母和数字出现 2 次?
32. 6 个人围着一个圆桌就座，如果通过旋转圆桌而使得就座方式从一种变成另一种，就认为它们是同一种方式，那么有多少种不同的就座方式?
33. 如果 n 和 k 是正整数，证明

$$C(n+1,k)=(n+1)C(n,k-1)/k$$

使用这个恒等式构造一个二项式系数的归纳定义。
34. 证明如果 P 是一个素数且 k 是一个整数，使得 $1\leqslant k\leqslant p-1$，则 p 整除 $C(p,k)$。
35. 求 $(x+y)^5$ 的展开式。
36. 求在 $(x+y)^{13}$ 的展开式中的 x^5y^8 的系数。
37. 在 $(x+y)^{100}$ 的展开式中有多少个项?
38. 在 $(1+x)^{11}$ 中 x^7 的系数是什么?
39. 在 $(2-x)^{19}$ 中 x^9 的系数是什么?
40. 在 $(3x+2y)^{17}$ 中 x^8y^9 的系数是什么?
41. 在 $(2x-3y)^{200}$ 中 $x^{101}y^{99}$ 的系数是什么?

*42. 给出一个关于 $(x+1/x)^{100}$ 的展开式中 x^k 系数的公式，其中 k 是一个整数。

*43. 给出一个关于 $(x^2-1/x)^{100}$ 的展开式中 x^k 系数的公式，其中 k 是一个整数。

44. 帕斯卡三角形中包含二项式系数 $C(10,k)$，$0\leqslant k\leqslant 10$ 的行是

1 10 45 120 210 252 210 120 45 10 1

用帕斯卡公式计算出在帕斯卡三角形中紧接这行下面的另一行。
45. 帕斯卡三角形中包含二项式系数 $C(9,k)(0\leqslant k\leqslant 9)$ 的行是什么?
46. 设 n 是正整数。什么是最大的二项式系数 $C(n,r)$? 这里的 r 是小于或等于 n 的非负整数。证明你的答案是正确的。
47. 使用关于 $C(n,r)$ 的公式证明帕斯卡恒等式。
48. 证明恒等式 $C(n,r)C(r,k)=C(n,k)C(n-k,r-k)$，其中 n，r 和 k 都是非负整数，满足 $r\leqslant n$ 和 $k\leqslant r$。

    a) 使用组合论证。

    b) 使用关于 n 元素集合的 r-组合数公式为基础来论证。

*49. 证明

$$\sum_{k=0}^{r}C(n+k,k)=C(n+r+1,r)$$

其中 n 和 r 是正整数。

a) 用组合论证。

b) 用帕斯卡恒等式。

50. 证明如果 n 是正整数，则 $C(2n,2)=2C(n,2)+n^2$。

a) 使用组合论证。

b）通过代数推导。

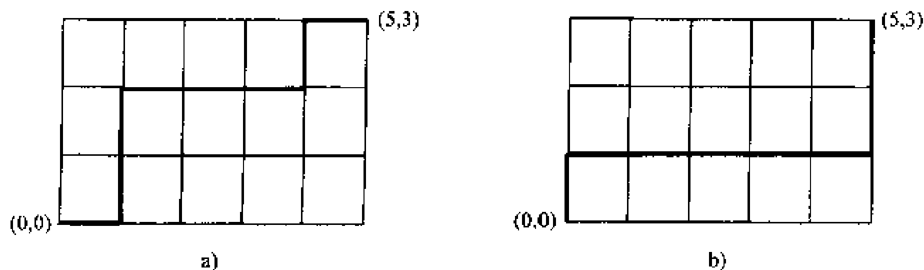
*51．给出关于$\sum_{k=1}^{n} C(n,k)=n2^{n-1}$的组合证明。[提示：以两种方法计数选择一个委员会然后选择这个委员会领导的方式数。]

*52．给出关于$\sum_{k=1}^{n} kC(n,k)^2=nC(2n-1,n-1)$的组合证明［提示：用两种方法计数选择一个委员会的方式数，如果这个委员会有 n 个成员，要求这些成员选自 n 个数学教授和 n 个计算机科学教授并使得委员会的主席是数学教授。]

53．证明集合具有奇数个元素的子集数与具有偶数个元素的子集数相等。

*54．使用数学归纳法证明二项式定理。

55．在这个练习里我们将要计数 xy 平面上在原点和(m,n)点之间的路径数。这些路径由一系列步构成，其中每一步是向右或者向上移动一个单位（不允许向左或向下移动）。下图给出了两条这种从$(0,0)$到$(5,3)$的路径。



a)　　　　b)

a）证明上述每条这种类型的路径可以用由 m 个 0 和 n 个 1 组成的二进制串表示，其中 0 表示向右移动一个单位，1 表示向上移动一个单位。

b）从 a）推断存在着$C(m+n,n)$条所求类型的路径。

56．用练习 55 证明$C(n,k)=C(n,n-k)$，其中 k 是一个整数，满足 $0\leqslant k\leqslant n$。[提示：考虑在练习 55 中所述的从$(0,0)$到$(n-k,k)$和从$(0,0)$到$(k,n-k)$的路径数。]

57．使用练习 55 证明定理 4。[提示：计数练习 55 所描述的那种 n 步路径数。每条路径必须在一个$(n-k,k)$点结束，其中 $k=0，1，2，\cdots，n$。]

58．使用练习 55 证明帕斯卡恒等式。［提示：显示一条在练习 55 所描述的那种从（0，0）到$(n+1-k,k)$并通过$(n+1-k,k-1)$点或$(n-1,k)$点但不同时通过这两点的路径。]

59．使用练习 55 证明练习 49 中的恒等式。[提示：首先注意到从$(0,0)$到$(n+1,r)$的路径数等于$C(n+1+r,r)$。其次，按照开始向上走 k 个单位分别计数每一类路径，其中 $k=0，1，2，\cdots，r$，然后对结果求和。]

*60．如果允许并列名次，4 匹马参加马赛有多少种比赛结果？（注意由于允许并列名次，4 匹马中多少匹并列都是可能的。）

*61．有 6 个人参加 100 码冲刺。如果允许并列名次，有多少种方式授予 3 块奖牌？（跑得最快的人得金牌，恰好只被一个人超过的人得银牌，恰好被两个人超过的人得铜牌。）

*62．为避免世界杯足球锦标赛总决赛中出现并列名次，通常采用下述过程。每个队按照预定的顺序选出 5 名球员。每个球员罚一个球，第一队的球员先罚，接着第二队的球员再罚，依照指定的顺序依次交替罚球。如果在 10 次罚球后得分还相等，再次重复这个过程。如果在 20 次罚球后得分仍旧相等，进行加赛时间的射门，第一个得分的队得胜。

a) 如果比赛进行第一轮的 10 个罚球，并且这轮比赛结束时一个队不可能与另一个队得分相等，那么有多少种不同的得分方案？

b) 如果比赛进行第二轮的 10 个罚球，对第一轮和第二轮罚球可能有多少种不同的得分方案？

c) 如果比赛在两轮每队罚 5 个球的加赛以后最多再射门 10 次，那么整个加赛过程可能有多少种得分方案？

*63. 如果一个序列的前若干项如下列出，对于它的第 n 项确定一个与二项式系数有关的公式。[提示：对帕斯卡三角形的观察有助于问题的求解。虽然以这一组给定的项作为开始的序列有无数多个，但下面列出的每个序列都是所求的那种序列的开始。]

a) 1，3，6，10，15，21，28，36，45，55，66，…

b) 1，4，10，20，35，56，84，120，165，220，…

c) 1，2，6，20，70，252，924，3432，12870，48620，…

d) 1，1，2，3，6，10，20，35，70，126，…

e) 1，1，1，3，1，5，15，35，1，9，…

f) 1，3，15，84，495，3003，18564，116280，735471，4686825，…

4.4 离散概率

4.4.1 引言

组合学与概率论有着共同的起源。概率论初始的发展是在 17 世纪，当时的法国数学家布莱斯·帕斯卡对某些赌博游戏进行了分析。就是在这些研究中帕斯卡发现了二项式系数的各种性质。在 18 世纪，法国数学家拉普拉斯[㊀]也研究赌博，并且把事件的概率定义为成功的结果除以可能的结果所得的商。例如，一个骰子掷出奇数点的概率就是成功结果的个数——即出现奇数点的个数除以可能结果的个数——即骰子可能出现的不同方式数。有 6 种可能的结果，即 1，2，3，4，5 和 6，其中恰好 3 种是成功的结果，即 1，3 和 5。因此，骰子掷出奇数点的概率是 3/6 = 1/2。（注意这里假定所有结果的可能性是相等的，或者换句话说，骰子是均匀的。）

这一节我们的讨论将局限于具有可能性相等的有限多个结果的实验。这将允许我们使用拉普拉斯关于事件概率的定义。我们将在 4.5 节继续研究概率，那里研究的是具有有限多个结果但结果的可能性不一定相等的实验。在 4.5 节我们也将引入概率论中的一些关键概念，包含条件概率、事件的独立性、随机变量以及期望值。

㊀ 皮埃尔·西蒙·拉普拉斯(Pierre Simon Laplace，1749—1827) 拉普拉斯出身于一个低下的诺曼底家族。他在童年于一所教会学校受到教育。16 岁他进入凯恩大学学习神学。但是不久他就意识到他真正感兴趣的是数学。在完成学业以后，他在凯恩大学担任临时教授。在 1769 年他成为巴黎陆军学校的数学教授。

拉普拉斯由于他对天体力学、天体运动研究所作出的贡献而闻名于世。他的《天体力学的特征》被认为是十九世纪初期最伟大的科学著作之一。拉普拉斯是概率论的奠基人之一，他也对数理统计学作出了许多项献。他把在这个领域的工作写成《概率论的理论分析》一书，书中一个事件的概率被定义为实验所希望的结果数与总结果数之比。

拉普拉斯也由于他的政治灵活性而著称。他先后忠实于法兰西共和国、拿破仑和路易十八皇帝。这使得他在法国大革命前、革命期间和革命后都成果卓著。

4.4.2 有限概率

一次实验是一个过程，这个过程将从一组可能的结果中得到一个结果。实验的样本空间是可能结果的集合。一个事件是样本空间的子集。现在叙述拉普拉斯关于具有有限多个可能结果的事件的概率定义。

定义1 一个事件 E 是具有相等可能性结果的有限样本空间 S 的子集，E 的概率是 $p(E)=|E|/|S|$。

下面再给出一些例子。

例1 一个缸里有4个蓝球和5个红球。从缸里取出一个蓝球的概率是多少？

解 为计算这个概率，首先考虑存在9个可能的结果，这些可能的结果中有4个得到蓝球。因此，取一个蓝球的概率是4/9。■

例2 掷两个骰子使得其点数之和等于7的概率是多少？

解 当掷两个骰子时总共有36种可能的结果。（这是由乘积法则得到的，因为每个骰子有6个可能的结果，当掷两个骰子时总共有 $6^2=36$ 种结果。）存在有6种成功的结果，即(1，6)，(2，5)，(3，4)，(4，3)，(5，2)和(6，1)，其中第一和第二个骰子的点数用一个有序对来表示。因此，当两个均匀的骰子被掷时，7出现的概率是6/36=1/6。■

目前彩票变得非常流行。我们可以轻松地算出赢各种不同类形彩票的机会。

例3 在一种彩票里，当人们挑的4个数字按照正确的次序与一个随机过程选出的4个数字匹配时他们就中了大奖。如果只有3个数字匹配，他们就中了比较小的奖。一个人赢大奖的概率是多少？赢小奖的概率是多少？

解 只有一种方式使得所选择的4个数字都正确。由乘积法则，有 $10^4=10000$ 种方式选四位数。因此，赢大奖的概率是1/10000=0.0001。

4个数字中恰好选对了3个数字的能够赢小奖。为了使得3个数字正确，而不是4个数字正确，必须恰好1个数字出错。可以先求选4个数字且除了第 i 个数字之外都与挑出的数字匹配的方式数，这里的 $i=1, 2, 3, 4$，然后对它们求和。根据求和法则，就能得到恰好选对3个数字的方式数。为计算使得第一个数字不匹配的选法数，注意对第一个数字有9种可能的选择（除了一个正确的数字外），而其他的每个数字只有一种选择，即对这些位置的正确数字。因此，选择4个数字使得第一个数字出错而后3个数字正确的有9种方式。类似地，有9种方式选择4个数字而使得第2个数字出错，9种方式使得第3个数字出错，9种方式使得第4个数字出错。从而总共有36种方式选择4个数字并恰好使得其中的3个是正确的。于是，一个人赢小奖的概率是36/10 000=9/2500=0.0036。■

例4 现在有许多彩票使那些从前 n 个正整数中选对6个数的人得到特别大奖，这里的 n 通常在30到50之间。一个人从40个数中选对6个数的概率是多少？

解 只有一个赢组合。从40个数中选6个数的总方法数是

$$C(40,6)=\frac{40!}{34!\ 6!}=3\ 838\ 380$$

因此，取出一个赢组合的概率是 1/3 838 380，近似等于 0.00000026。■

我们可以使用目前开发的技术找到扑克游戏中某些牌的概率。一副牌有 52 张，分成 13 类，每类 4 张。这些类是 2，3，4，5，6，7，8，9，10，J，Q，K 和 A。还有 4 套花色，即黑桃、梅花、红心和方块，每套包含 13 张，并且每套里每类只有 1 张牌。

例 5　如果 5 张为一手牌，从 52 张牌中选一手牌有多少种不同的牌？

解　若 5 张为一手牌，存在 $C(52,5)=2\ 598\ 960$ 种不同的牌。■

例 6　一手扑克牌有 5 张，如果其中的 4 张牌是同一类的，找出这种牌的概率。

解　由乘积法则，5 张牌中 4 张牌是同类的数目就是选取一类的方式数，从该类的 4 张牌中选取 4 张牌的方式数与选第 5 张牌的方式数之积，即

$$C(13,1)C(4,4)C(48,1)$$

由于 5 张牌总共存在 $C(52,5)$ 种，一手牌中包含 4 张同类牌的概率是

$$\frac{C(13,1)C(4,4)C(48,1)}{C(52,2)}=\frac{13\cdot 1\cdot 48}{2\ 598\ 960}$$（近似等于 0.00024）■

例 7　一手牌包含一个完全的族，即 3 张在同一类且其余 2 张在另一类的概率是多少？

解　由乘积法则，包含一个完全的族的牌数是有序的选取两个类的方式数，第一类的 4 张牌选 3 张的方式数和第二类的 4 张牌选 2 张的方式数之积。（注意两类的次序是有关的，例如 3 张 Q 和 2 张 A，与 3 张 A 和 2 张 Q 是不同的。）可以看出包含一个完全的族的牌数是

$$P(13,2)C(4,3)C(4,2)=13\cdot 12\cdot 4\cdot 6=3744$$

因为存在 2 598 960 手牌，一个完全的族的概率是

$$\frac{3\ 744}{2\ 598\ 960}$$（近似等于 0.0014）■

4.4.3　事件组合的概率

我们可以使用计数技术得到从其他事件导出事件的概率。

定理 1　设 E 是样本空间 S 的一个事件。事件 $\overline{E}$（事件 E 的补事件）的概率是

$$p(\overline{E})=1-p(E)$$

证　为求出事件 $\overline{E}$ 概率，注意 $|\overline{E}|=|S|-|E|$。因此，

$$p(\overline{E})=\frac{|S|-|E|}{|S|}=1-\frac{|E|}{|S|}=1-p(E)$$ ■

当直接的方法不适用时可以用另一种策略来找事件的概率。不用确定这个事件的概率，但可以确定它的补事件的概率。这常常是更容易做到的，正如下面的例子所显示的。

例 8　随机生成一个 10 位二进制序列，其中至少 1 位是 0 的概率是多少？

解　设 E 是 10 位中至少一位是 0 的事件。那么 $\overline{E}$ 是所有的位都是 1 的事件。因为样本

空间是所有 10 位二进制串的集合，从而得到

$$\begin{aligned} p(E) &= 1 - p(\overline{E}) \\ &= 1 - \frac{|\overline{E}|}{|S|} \\ &= 1 - \frac{1}{2^{10}} \\ &= 1 - \frac{1}{1024} \\ &= \frac{1023}{1024} \end{aligned}$$

所以，包含至少一位 0 的二进制串的概率是 1023/1024。直接使用定理 1 找到这个概率是相当困难的。 ■

我们也可以求出两个事件的并集的概率。

定理 2　设 E_1 和 E_2 是样本空间的事件，那么

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$$

证　使用 1.4 节给出的关于两个集合的并集的元素数公式得到

$$|E_1 \cup E_2| = |E_1| + |E_2| - |E_1 \cap E_2|$$

因此，

$$\begin{aligned} p(E_1 \cup E_2) &= \frac{|E_1 \cup E_2|}{|S|} \\ &= \frac{|E_1| + |E_2| - |E_1 \cap E_2|}{|S|} \\ &= \frac{|E_1|}{|S|} + \frac{|E_2|}{|S|} - \frac{|E_1 \cap E_2|}{|S|} \\ &= p(E_1) + p(E_2) - p(E_1 \cap E_2) \end{aligned}$$

□

例 9　随机从一组不超过 100 的正整数中选出一个正整数使得它能被 2 或 5 整除的概率是多少？

解　设 E_1 是选出一个能被 2 整除的数的事件，E_2 是选出一个能被 5 整除的数的事件。那么 $E_1 \cup E_2$ 是能被 2 或 5 整除的事件，$E_1 \cap E_2$ 是能被 2 和 5 同时整除的事件，换句话说，即能被 10 整除的事件。由于 $|E_1| = 50$，$|E_2| = |20|$，且 $|E_1 \cap E_2| = 10$，从而得到

$$\begin{aligned} p(E_1 \cup E_2) &= p(E_1) + p(E_2) - p(E_1 \cap E_2) \\ &= \frac{50}{100} + \frac{20}{100} + \frac{10}{100} \\ &= \frac{3}{5} \end{aligned}$$

■

4.4.4　概率的推理

一个通常的问题是确定两个事件中哪一个更有可能发生？分析这些事件的概率可能是复杂的。下面的例子描述了一个这种类型的问题。它讨论了一个起源于电视剧“让我们做一次

交易”的著名问题。

例 10 Monty 大厦的 3 门难题。假定你是一个游戏节目的竞争者，你有机会赢一个大奖。要求你从 3 扇门中选一个打开，大奖就在其中某扇门的后面。一旦你选中了某扇门，游戏节目主持人知道每扇门后面是什么，他就如下进行。首先，不管你是否选择了赢大奖的门，他打开另外两扇门中的一扇没有奖的门（如果两扇门都没有奖，随机选择一扇）。然后他问你是否愿意换另外一扇门。你应该用什么策略？你应该换一扇门还是坚持原来的选择，或者这无关紧要？

解 你选对了门的概率（在主持人打开一扇门并且问你是否想改变之前）是 1/3，因为这 3 扇门成为正确的门的可能性相等。一旦游戏节目主持人打开另外两扇门中的一扇，这是正确的门的概率不变，因为他总是打开后面没有大奖的门。

你选错门的概率就是大奖在你没有选的两扇门中某一扇后面的概率。因此，你选错了门的概率是 2/3。如果你选错了，那么当游戏节目主持人打开一扇门向你显示大奖不在它的后面时，大奖一定在另一扇门的后面。若你原来的选择是错的并且改变了门，那么你总能赢。因此，通过改变门，你赢的概率是 2/3。换句话说，当节目主持人给你这样做的机会时，你总应该改变门，这使得你赢的概率增加了一倍。 ■

练习

1. 从一副牌中选出 1 个 A 的概率是多少？
2. 掷骰子时出现 6 点的概率是多少？
3. 从前 100 个正整数中随机选出 1 个奇数的概率是多少？
4. 从一年（366 天）中随机选出 1 天在 4 月的概率是多少？
5. 当掷 2 个骰子时，其点数之和是偶数的概率是多少？
6. 从一副牌中选 1 张牌，使得它是 1 张 A 或 1 张红心的概率是多少？
7. 掷 6 次硬币全部头像向上的概率是多少？
8. 一手扑克牌有 5 张，其中包含红心 A 的概率是多少？
9. 一手扑克牌有 5 张，其中不包含红心 Q 的概率是多少？
10. 一手扑克牌有 5 张，其中包含方块 2 和黑桃 3 的概率是多少？
11. 一手扑克牌有 5 张，其中包含方块 2、黑桃 3、红心 6、梅花 10 和红心 K 的概率是多少？
12. 一手扑克牌有 5 张，其中恰好包含 1 张 A 的概率是多少？
13. 一手扑克牌有 5 张，其中至少包含 1 张 A 的概率是多少？
14. 一手扑克牌有 5 张，其中包含 5 类不同的牌的概率是多少？
15. 一手扑克牌有 5 张，其中包含 2 个对子（两类不同的牌中每类有 2 张、第三类牌是第 5 张）的概率是多少？
16. 一手扑克牌有 5 张，其中包含一手同花，即 5 张牌的花色相同的概率是多少？
17. 一手扑克牌有 5 张，其中包含一个顺子，即 5 张牌的类是连续的概率是多少？（注意可以将 A 看作顺子 A-2-3-4-5 中的最低牌，也可以看作顺子 10-J-Q-K-A 中的最高牌。）
18. 一手扑克牌有 5 张，其中包含一个同花顺子，即 5 张牌的类是连续的也是同一花色的概率是多少？

*19．一手扑克牌有 5 张，其中包含 5 张不同类的牌且不包含一个同花或一个顺子的概率是多少？

20．一手扑克牌有 5 张，其中包含一个皇家的同花，即同一花色的 10，J，Q，K 和 A 的概率是多少？

21．一个骰子掷 6 次不出现偶数点的概率是多少？

22．随机选取一个不超过 100 的正整数能够被 3 整除的概率是多少？

23．随机选取一个不超过 100 的正整数能够被 5 或 7 整除的概率是多少？

24．求从不超过下述整数的正整数中选对 6 个整数来赢彩票的概率，这里不管选择整数的顺序。

a) 30　　b) 36　　c) 42　　d) 48

25．求从不超过下述整数的正整数中选中 6 个整数来赢彩票的概率，这里不管选择整数的顺序。

a) 50　　b) 52　　c) 56　　d) 60

26．求从不超过下述整数的正整数中选错 6 个整数的概率，这里不管选择整数的顺序。

a) 40　　b) 48　　c) 56　　d) 64

27．求从不超过下述整数的正整数中选 6 个整数，并且恰好选对 1 个的概率，这里不管选择整数的顺序。

a) 40　　b) 48　　c) 56　　d) 64

28．在宾夕法尼亚超级彩票中，买彩票的人要从前 80 个正整数中选出 7 个数。如果这 7 个数是在由宾夕法尼亚彩票委员会选出的 11 个数之中就能赢大奖，那么一个人赢大奖的概率是多少？

29．在一种超级彩票中，如果买彩票的人选中的 8 个数正是计算机从不超过 100 的正整数中选出的数就能中彩。买彩票的人赢这种超级彩票的概率是多少？

30．由计算机从 1 到 40 之间（包括 1 和 40 在内）选出 6 个数，如果某人选中了其中的 5 个（但不是 6 个）数就能获奖，那么获奖的概率是多少？

31．在轮盘赌中，旋转一个有 38 个数的轮盘，其中 18 个数是红的，18 个数是黑的，另外两个既不红也不黑的数是 0 和 00。当轮盘转动时它停在任何特定数字的可能性是 1/38。

a) 轮盘停在 1 个红数的概率是多少？

b) 轮盘旋转 2 次，2 次都停在 1 个黑数的概率是多少？

c) 轮盘停在 0 或 00 的概率是多少？

d) 轮盘旋转 5 次，5 次都不停在 0 或 00 的概率是多少？

e) 某次转动轮盘停在 1 和 6 之间（包含 1 和 6 在内）的某个数字，但下次转动轮盘不停在这些数字之间的概率是多少？

32．掷 2 个骰子总点数为 8 或掷 3 个骰子总点数为 8，哪种可能性更大？

33．掷 2 个骰子总点数为 9 或掷 3 个骰子总点数为 9，哪种可能性更大？

34．设 E_1 和 E_2 是两个事件，如果 $p(E_1 \cap E_2) = p(E_1)p(E_2)$，就称 E_1 和 E_2 是独立的。当一枚硬币被抛掷 3 次时所有可能的结果构成一个集合，如果把这个集合的子集看作事件，确定下面的每一对事件是否是独立的。

a) E_1：第一次硬币头像向下；E_2：第二次硬币头像向上。

b) E_1：第一次硬币头像向下；E_2：在连续3次中有2次但不是3次头像向上。

c) E_1：第二次硬币头像向下；E_2：在连续3次中有2次但不是3次头像向上。

（我们将在4.5节更深入地研究事件的独立性。）

35. 解释下面的句子错在什么地方。在Monty大厦三门难题里，因为剩下了两个门，你选的第一个门后面是大奖的概率与没有打开的另两个门后面是大奖的概率都是1/2。

36. 假定在Monty大厦难题中不是3个门而是4个门。当知道每个门后面是什么的主持人打开一个错门并且给你机会改变选择时，你不做改变并且赢了大奖的概率是多少？在你没有选的3个门剩下2个时，你把原来选的门改为这两个门之一并且赢得大奖的概率是多少？

4.5 概率论

4.5.1 引言

在节4.4我们引入了事件概率的概念。（回忆一下，一个事件是一次实验的可能结果的子集。）我们按照拉普拉斯的记号定义事件 E 的概率

$$p(E)=\frac{|E|}{|S|}$$

即 E 中的结果数除以结果总数。这个定义假定所有结果的可能性都是相等的。但是许多实验结果的可能性并不相等。例如，一个硬币很可能是不均匀的，因而出现头像向上的次数常常是向下次数的两倍。类似地，一个线性搜索的输入是一个元素和一个表，这个元素在表里或不在表里的可能性依赖于输入是怎么产生的。在这种情况下怎样建立关于事件可能性的模型呢？这一节我们将要说明当结果的可能性不相等时为研究实验概率应该怎样定义结果的概率。

假定一个均匀的硬币被掷4次，第一次它的头像向上。给定了这个信息，头像3次向上的概率是什么？为了回答这个或者类似的问题，我们将引入条件概率的概念。已知第一次头像向上能改变3次头像向上的概率吗？如果不是，这两个事件就叫做独立的，本节的后面将要学到这个概念。

许多问题谈到一个与实验结果有关的特定数值。例如，当我们掷100次硬币时，恰好出现40次头像的概率是多少？我们应该预期出现多少次头像？在这一节我们将要学习随机变量，它是把数值与实验结果联系起来的函数，而它们的加权平均叫做期望值。

4.5.2 概率赋值

设 S 是某个具有有穷个或可数个结果的实验的样本空间。我们对每个结果 s 赋予一个概率 $p(s)$，使得满足以下两个条件：

(i) $0\leqslant p(s)\leqslant 1$，对每个 $s\in S$

(ii) $\sum_{s\in S}p(s)=1$

条件（i）说明每个结果的概率是一个不超过1的非负实数。条件（ii）说明所有可能结果的概率之和应该是1，即当我们做这个实验时这些结果之一一定出现。这是拉普拉斯定义的一

般化。在拉普拉斯定义中对 n 个结果中的每一个都赋给 $1/n$ 的概率。的确，当使用拉普拉斯关于结果的可能性相同的概率定义时条件(i)和(ii)是满足的（见练习 4）。

注意到当在 n 个可能的结果 x_1，x_2，…，x_n 时，这两个要满足的条件是

(i) $0 \leqslant p(x_i) \leqslant 1$，对 $i = 1, 2, \cdots, n$

和

(ii) $\sum_{i=1}^{n} p(x_i) = 1$

为了建立实验的模型，对结果 s 赋的概率 $p(s)$ 应该等于 s 出现次数除以实验进行的次数。当这个数无限增加时，就取极限。(我们将假定讨论的所有实验有着平均可预料的结果，以使得这个极限存在。我们也假定一个实验的结果成功与否与前面的结果无关。)

注意 在这一节我们将要求可能的结果数是有限的。使用无限序列可以类似地处理可数的无限个结果，正如节末的练习 37-40 所显示的。我们将不讨论结果集合不是离散时的事件概率，例如当一个事件的结果可能是任何实数时。在这种情况下，对于事件概率的研究通常要求微积分。

我们可以建立实验的模型，在这种实验中结果具有相同的可能性，或者不相同但可以选择一个适当的函数 $p(s)$，正如例 1 所示。

例 1 当一个均匀的硬币被掷时，结果 H（头像向上）和结果 T（头像向下）应该赋给什么概率？当硬币不均匀而使得出现头像向上的次数常常是向下的两倍，对这些事件又应该赋予什么概率？

解 对于均匀的硬币，当硬币被掷时头像向上的概率等于头像向下的概率，这两个事件具有相同的可能性。因此，我们对这两个结果中的任何一个都赋给 1/2 的概率，即 $p(H) = p(T) = 1/2$。

对于不均匀的硬币我们有

$$p(H) = 2p(T)$$

由于

$$p(H) + p(T) = 1$$

从而得出

$$2p(T) + p(T) = 3p(T) = 1$$

最终有 $p(T) = 1/3$ 和 $p(H) = 2/3$。 ■

现在我们把事件的概率定义成在这个事件中结果的概率之和。

定义 1 事件 E 的概率是在 E 中结果的概率之和，即

$$p(E) = \sum_{s \in E} p(s)$$

注意当事件 E 中有 n 个结果时，即如果 $E = \{a_1, a_2, \cdots, a_n\}$，则 $p(E) = \sum_{i=1}^{n} p(a_i)$。

例 2 假定一个骰子是不均匀的（或经装填的）使得 3 出现的次数是其他数字的两倍，

但其他 5 个数出现的可能性相等。当我们掷这个骰子时出现奇数的概率是多少？

解 我们想要找到事件$E=\{1,3,5\}$的概率。由本节末的练习 2，我们有

$$p(1)=p(2)=p(4)=p(5)=p(6)=1/7, p(3)=2/7$$

从而得出

$$p(E)=p(1)+p(3)+p(5)=1/7+2/7+1/7=4/7$$ ■

当事件是等可能的并且存在有限多个可能的结果时，在这一节给出的事件概率的定义（定义 1）与拉普拉斯的定义（4.4 节定义 1）一致。为此，假定存在 n 个等可能的结果；由于这些概率之和是 1，因此每个可能结果的概率是 $1/n$。假定事件 E 包含 m 个结果，根据定义 1，

$$p(E)=\sum_{i=1}^{m}\frac{1}{n}=\frac{m}{n}$$

由于$|E|=m$ 和$|S|=n$，从而有

$$p(E)=\frac{m}{n}=\frac{|E|}{|S|}$$

这是事件 E 的拉普拉斯的概率定义。

4.5.3 事件的组合

当我们使用定义 1 来定义事件概率时，在 4.4 节中事件组合的概率公式继续保持。例如，4.4 节定理 1 断言

$$p(\overline{E})=1-p(E)$$

其中 $\overline{E}$ 是事件E 的补事件。当用定义 1 时这个等式也成立。为此只需注意到 n 个可能结果的概率之和是 1，且每个结果或在 E 或在$\overline{E}$ 中，但不能同时在两者之中。因而

$$\sum_{s\in S}p(s)=1=p(E)+(\overline{E})$$

所以，$p(\overline{E})=1-p(E)$。

根据拉普拉斯的定义，由 4.4 节定理 2，我们有

$$p(E_1\cup E_2)=p(E_1)+p(E_2)-p(E_1\cap E_2)$$

其中 E_1 和 E_2 是样本空间 S 的事件。当我们按照这一节的作法定义事件的概率时等式也成立。为此，注意到$p(E_1\cup E_2)$是在 $E_1\cup E_2$ 中结果的概率之和。当结果 x 只属于E_1 和 E_2 中的一个集合但不同时属于两个集合时，$p(x)$恰好只出现在$p(E_1)$或$p(E_2)$的一个和里。当结果 x 同时在E_1 和 E_2 中时，$p(x)$出现在$p(E_1)$的和里，$p(E_2)$的和里，也出现在$p(E_1\cap E_2)$的和里。因此它在右边出现了 $1+1-1=1$ 次。所以，左边与右边相等。

4.5.4 条件概率

假定我们掷 3 次硬币并且所有的 8 种可能都是等可能的。此外，假定我们知道第一次掷硬币头像向下的事件 F 已经出现了。在给定这一信息后，事件 E，即头像向下出现奇数次

的概率是什么？因为第一次掷硬币的头像向下，只有 4 种可能的结果：TTT，TTH，THT 和 THH，其中 H 和 T 分别表示头像向上和向下。头像向下出现奇数次只有结果 TTT 和 THH。由于 8 个结果的概率相等，在给定 F 出现的条件下，4 种可能的结果的每一个也应该有相等的概率 1/4。这就告诉我们，在给定 F 出现的条件下应该对 E 的概率赋值 $2/4 = 1/2$。这个概率叫做给定 F 的条件下 E 的条件概率。

一般说来，为了找出给定 F 的条件下 E 的条件概率，我们用 F 作为样本空间。作为要出现的 E 的一个结果，这个结果也必须属于 $E \cap F$。出于这一考虑，我们得到下述定义。

定义 2 设 E 和 F 是具有 $p(F) > 0$ 的事件。给定 F 的条件下 E 的条件概率记作 $p(E|F)$，定义为

$$p(E|F) = \frac{p(E \cap F)}{p(F)}$$

例 3 随机生成 4 位的二进制串以使得 16 个位串都是等可能的，那么在给定串的第一位是 0 的条件下，包含至少两个连续 0 的串的概率是多少？（我们假定 0 位和 1 位是等可能的。）

解 设 E 是包含至少 2 个连续 0 的 4 位二进制串的事件，F 是 4 位二进制串的第一位是 0 的事件。那么在给定第一位是 0 的条件下，包含至少 2 个连续 0 的 4 位二进制串的概率是

$$p(E|F) = \frac{p(E \cap F)}{p(F)}$$

由于 $E \cap F = \{0000, 0001, 0010, 0011, 0100\}$，故 $p(E \cap F) = 5/16$。因为以 0 开始的 4 位二进制串有 8 个，因而 $p(F) = 8/16 = 1/2$。所以，

$$p(E|F) = \frac{5/16}{1/2} = \frac{5}{8}$$ ■

例 4 在至少有 1 个男孩的条件下，一个有 2 个孩子的家庭有 2 个男孩的条件概率是多少？假定 BB，BG，GB 和 GG 是等可能的，其中 B 代表男孩，G 代表女孩。

解 设 E 是有 2 个孩子的家庭有 2 个男孩的事件，F 是一个有 2 个孩子的家庭至少有 1 个男孩的事件。因而 $E = \{BB\}$，$F = \{BB, BG, GB\}$，并且 $E \cap F = \{BB\}$。由于 4 种可能性是等可能的，故 $p(F) = 3/4$ 且 $p(E \cap F) = 1/4$。从而可以断言

$$p(E|F) = \frac{p(E \cap F)}{p(F)} = \frac{1/4}{3/4} = \frac{1}{3}$$ ■

4.5.5 独立性

假设一个硬币被掷了 4 次，正如我们关于条件概率讨论的引言中所描述的。是否知道第一次掷出的头像向下（事件 F）改变了奇数次头像向下（事件 E）的概率？换句话说，$p(E|F) = p(E)$？由于 $p(E|F) = 1/2$ 和 $p(E) = 1/2$，这个等式对事件 E 和 F 是有效的。因为这个等式成立，我们说 E 和 F 是**独立事件**。

由于 $p(E|F) = p(E \cap F)/p(F)$，问是否 $p(E|F) = p(E)$ 与问是否 $p(E \cap F) =$

$p(E)p(F)$是一样的。从而得到下面的定义。

定义 3 事件 E 和 F 是独立的，当且仅当$p(E\cap F)=p(E)p(F)$。

例 5 假设 E 是随机产生以一个 1 开始的 4 位二进制串的事件，F 是随机产生包含偶数个 0 的二进制串的事件。如果 16 个 4 位二进制串是等可能的，E 和 F 是独立的吗？

解 以 1 开始的 4 位二进制串有 8 个:1000,1001,1010,1011,1100,1101,1110 和 1111。包含偶数个 0 的 4 位二进制串也有 8 个：0000，0011，0101，0110，1001，1010，1100，1111。因为 4 位二进制串有 16 个，故

$$p(E)=p(F)=8/16=1/2$$

由于$E\cap F=\{1111,1100,1010,1001\}$,因此

$$p(E\cap F)=4/16=1/4$$

因为

$$p(E\cap F)=1/4=(1/2)(1/2)=p(E)p(F)$$

我们断定 E 和 F 是独立的。 ■

例 6 和例 4 类似，假定一个家庭可以有两个孩子的 4 种情况是等可能的。事件 E 是有两个孩子的家庭有两个男孩，事件 F 是有两个孩子的家庭至少有一个男孩，E 和 F 是否独立？

解 因为$E=\{BB\}$,我们有$p(E)=1/4$。在例4 我们证明了$p(F)=3/4$ 和$p(E\cap F)=1/4$。由于$p(E\cap F)=1/4\neq 3/16=(1/4)(3/4)=p(E)p(F)$,事件 E 和 F 不是独立的。 ■

例 7 事件 E 是某个有三个孩子的家庭有男孩也有女孩，F 是有三个孩子的家庭有至多有一个男孩。假定一个家庭可能有三个孩子的 8 种方式是等可能的，E 和 F 是否独立？

解 一个家庭可能有三个孩子的 8 种方式是$BBB,BBG,BGB,BGG,GBB,GBG,GGB,GGG$,每一种的概率都是 1/8。因为$E=\{BBG,BGB,BGG,GBB,GBG,GGB\}$,$F=\{BGG,GBG,GGB,GGG\}$,并且$E\cap F=(BGG,GBG,GGB)$,从而$p(E)=6/8=3/4$,$p(F)=4/8=1/2$,且 $p(E\cap F)=3/8$。由于

$$p(E\cap F)=\frac{3}{8}=\frac{3}{4}\cdot\frac{1}{2}=p(E)p(F)$$

可以断言 E 和 F 是独立的。(这个结论似乎是令人惊奇的。的确，如果我们改变孩子数，结论可能不再成立。见节末的练习 19。) ■

4.5.6 伯努利实验与二项式分布

假设一个实验只有两种可能的结果。例如，当随机产生一个二进制位时，可能的结果就是 0 和 1。当一个硬币被掷时，可能的结果就是头像向上和头像向下。每次实行一

项具有两种可能结果的实验就叫做一次伯努利实验。它是以詹姆斯·伯努利[⊖]命名的，他对概率论作出了重要的贡献。一般地说，一次伯努利实验的一个可能的结果叫做*成功*或*失败*。如果 p 是一次成功的概率，q 是一次失败的概率，那么 $p+q=1$。

当一个实验由 n 次独立的伯努利实验组成时，许多问题可以通过确定 k 次成功的概率来解决。考虑下面的例子。

例 8　一枚硬币是不均匀的，以至于出现头像的概率是 2/3。假定每次掷硬币是独立的，当掷 7 次硬币时恰好 4 次出现头像的概率是多少?

解　当一枚硬币被掷 7 次时存在 $2^7=128$ 种可能的结果，7 次中有 4 次出现头像的方式数是 $C(7,4)$。因为 7 次掷币是独立的，每一个这样的结果都有概率 $(2/3)^4(1/3)^3$。因此，恰好 4 次出现头像的概率是

$$C(7,4)(2/3)^4(1/3)^3=\frac{35.16}{3^7}=\frac{560}{2187}$$

■

参照在例 8 中使用的推理，我们可以建立下面的定理，它告诉我们在 n 次独立的伯努利实验中有 k 次成功的概率。

定理 1　*在 n 次独立的伯努利实验中有 k 次成功的概率*在成功概率为 p，失败概率为 q 的 n 次独立的伯努利实验中有 k 次成功的概率是

$$C(n,k)p^kq^{n-k}$$

证　当执行 n 次伯努利实验时，结果是 n 元组 $(t_1,t_2,\cdots,t_n)$，其中 $t_i=S$（成功）或 $t_i=F$（失败），$i=1,2,\cdots,n$。由于 n 次实验是独立的，由 k 次成功和 $n-k$ 次失败（以任何顺序）组成的每个 n 次实验结果的概率是 p^kq^{n-k}。因为由 S 和 F 构成的包含 k 个 S 的 n 元组有 $C(n,k)$ 个，k 次成功的概率是

$$C(n,k)p^kq^{n-k}$$

□

我们将成功概率为 p、失败概率为 $q=1-p$ 的 n 次独立的伯努利实验中有 k 次成功的概率记作 $b(k;n,p)$。作为 k 的函数，我们把这个函数称为*二项式分布*。定理 1 告诉我们，$b(k;n,p)=C(n,k)p^kq^{n-k}$。

例 9　当产生 10 位二进制串时，若产生一位 0 的概率是 0.9，一位 1 的概率是 0.1，且每一位的产生是独立的，那么产生恰好 8 位 0 的概率是多少?

⊖ 詹姆斯·伯努利(James Bernoulli, 1654—1705)　伯努利又名雅各布，诞生在瑞士的巴塞尔。他是伯努利家族的八位卓越的数学家之一（见 8.1 节数学家的伯努利家谱）。遵从他父亲的意愿，詹姆斯学习了神学并且担任了神职。但和他父亲的愿望相反，他也研究数学和天文学。从 1676 到 1682 年他遍及欧洲旅行，获悉了数学和科学的最新发现。他在 1682 年返回巴塞尔，创立了数学和科学学校。1687 年他被任命为巴塞尔大学的数学教授，并在这个位置终其一生。

詹姆斯·伯努利最知名的著作是《*Ars Conectandi*》，发表在他死后 8 年。在这本著作中他描述了在概率论和枚举中的已知结果，并常常对已知结果提供另外的证明。这本著作也包含了概率论对机会对策的应用和关于著名的大数定理的介绍。这条定理叙述了如果 $\epsilon>0$，当 n 变得任意大时，事件 E 在 n 次实验中出现的次数除以 n 的比与 $p(E)$ 的差在 ϵ 之内的概率接近于 1。

解 由定理1，恰好产生8位0的概率是

$$b(8;10,0.9)=C(10,8)(0.9)^8(0.1)^2=0.1937102445$$ ■

注意当执行 n 次独立的伯努利实验时，对于 $k=0, 1, 2, \cdots, n$，存在 k 次成功的概率之和等于

$$\sum_{k=0}^{n} C(n,k)p^k q^{n-k}=(p+q)^n=1$$

显然应该如此。在这串等式中的第一个相等是二项式定理的结果，第二个相等是由于 $q=1-p$。

4.5.7 随机变量

许多问题都涉及到一个与实验结果相关的数值。例如，我们可能想知道当随机产生10位二进制串时含9个1的概率，或者我们想知道掷20次硬币时有11次头像向下的概率。为了研究这类问题我们引入随机变量的概念。

定义4 一个随机变量是从实验的样本空间到实数集的函数。即是说，一个随机变量对每个可能的结果赋一个实数值。

注意 一个随机变量是一个函数，而不是一个变量，并且它也不是随机的！

例10 假设一个硬币被掷3次。令 $X(t)$ 是出现头像的个数，其中 t 是结果。那么随机变量 $X(t)$ 取值如下：

$$X(HHH)=3$$
$$X(HHT)=X(HTH)=X(THH)=2$$
$$X(TTH)=X(THT)=X(HTT)=1$$
$$X(TTT)=0$$ ■

例11 设 X 是掷一对骰子时的出现的点数之和。那么这个随机变量对36个可能的结果 (i,j) 取什么值？这里的 i 和 j 分别表示当掷两个骰子时第一个和第二个骰子出现的点数。

解 随机变量 X 取值如下：

$$X((1,1))=2$$
$$X((1,2))=X((2,1))=3$$
$$X((1,3))=X((2,2))X((3,1))=4$$
$$X((1,4))=X((2,3))=X((3,2))=X((4,1))=5$$
$$X((1,5))=X((2,4))=X((3,3))=X((4,2))=X((5,1))=6$$
$$X((1,6))=X((2,5))=X((3,4))=X((4,3))=X((5,2))=X((6,1))=7$$
$$X((2,6))=X((3,5))=X((4,4))=X((5,3))=X((6,2))=8$$
$$X((3,6))=X((4,5))=X((5,4))=X((6,3))=9$$
$$X((4,6))=X((5,5))=X((6,4))=10$$
$$X((5,6))=X((6,5))=11$$
$$X((6,6))=12$$ ■

4.5.8　期望值

许多问题可以用我们所期望的随机变量的取值，或者更精确地说是用随机变量在大量实验中的平均值来表示。这类问题包含：当掷 100 次硬币时预期会出现多少次头像？在表中线性查找一个元素时预期的比较次数是多少？为研究这类问题我们引入关于一个随机变量的期望值的概念。

定义 5　随机变量 $X(s)$ 在样本空间 S 的期望值（或期望）等于

$$E(X)=\sum_{s\in S}p(s)X(s)$$

注意当样本空间 S 有 n 个元素时，$S=\{x_1,x_2,\cdots,x_n\}$，$E(X)=\sum_{i=1}^{n}p(x_i)X(x_i)$。

注意　这里我们仅关心与有限期望值相关的随机变量。

例 12　一个均匀的硬币被掷了 3 次。令 S 是 8 种可能结果的样本空间，X 是随机变量，它对结果的赋值是结果中的头像数。那么 X 的期望值是什么？

解　在例 10 中我们列出了掷 3 次硬币时 X 对 8 个可能结果的值。由于硬币是均匀的且每次掷硬币是独立的，每个结果的概率都是 1/8。因此，

$$\begin{aligned}E(X)&=\frac{1}{8}(X(HHH)+X(HHT)+X(HTH)+X(THH)+X(TTH)\\&\quad+X(THT)+X(HTT)+X(TTT))\\&=\frac{1}{8}(3+2+2+2+1+1+1+0)\\&=\frac{12}{8}\\&=\frac{3}{2}\end{aligned}$$

■

当一个实验有相对较少的结果时，我们可以直接从定义计算随机变量的期望值，正像在例 12 中所做的。但是，当一个实验有许多结果时，直接由定义计算随机变量的期望值可能是不方便的。换一种作法，我们可以将随机变量的值相等的实验结果分成组来找随机变量的期望值。特别地，假设 X 是值域为 $X(S)$ 的随机变量，$p(X=r)$ 是随机变量 X 取值 r 的概率。因此，$p(X=r)$ 是使得 $X(s)=r$ 的结果 s 的概率。从而得到

$$E(X)=\sum_{r\in X(S)}p(X=r)r$$

例 13 和 14 说明了这个公式的用法。在例 13 中我们将找出掷两个均匀的骰子出现的点数之和的期望值。在例 14 中，我们将找出当执行 n 次伯努利实验时成功次数的期望值。

例 13　当掷一对均匀的骰子时所出现的点数之和的期望值是什么？

解　设 X 是随机变量，它等于掷一对骰子所出现的点数之和。在例 11 我们列出了这个实验的 36 个结果的值。X 的值域是 $\{2,3,4,5,6,7,8,9,10,11,12\}$。由例 11 我们有

$$p(X=2)=p(X=12)=1/36$$
$$p(X=3)=p(X=11)=2/36=1/18$$

$$p(X=4)=p(X=10)=3/36=1/12$$
$$p(X=5)=p(X=9)=4/36=1/9$$
$$p(X=6)=p(X=8)=5/36$$
$$p(X=7)=6/36=1/6$$

把这些值带入公式，得

$$\begin{aligned}E(X)&=2\cdot\frac{1}{36}+3\cdot\frac{1}{18}+4\cdot\frac{1}{12}+5\cdot\frac{1}{9}+6\cdot\frac{5}{36}+7\cdot\frac{1}{6}\\&\quad+8\cdot\frac{5}{36}+9\cdot\frac{1}{9}+10\cdot\frac{1}{12}+11\cdot\frac{1}{18}+12\cdot\frac{1}{36}\\&=7\end{aligned}$$

■

例 14 当执行 n 次伯努利实验时成功次数的期望值是什么？这里 p 是每次实验成功的概率。

解 令 X 是等于 n 次实验中成功次数的随机变量。由定理 1 我们看到 $p(X=k)=C(n,k)p^kq^{n-k}$。于是把随机变量赋值相同的结果分到一组，并根据分组来求随机变量的期望值的公式，我们有

$$\begin{aligned}E(X)&=\sum_{k=1}^{n}kp(X=k)\\&=\sum_{k=1}^{n}kC(n,k)p^kq^{n-k}\\&=\sum_{k=1}^{n}nC(n-1,k-1)p^kq^{n-k}\\&=np\sum_{k=1}^{n}C(n-1,k-1)p^{k-1}q^{n-k}\\&=np\sum_{j=0}^{n-1}C(n-1,j)p^jq^{n-1-j}\\&=np(p+q)^{n-1}\\&=np\end{aligned}$$

证明的第三个等式是由 4.3 节的练习 33 导出的 $C(n,k)=nC(n-1,k-1)/k$ 得到的。第五个等式是将 $j=k-1$ 代入求和的序标使得当 k 从 1 变到 n 时 j 从 0 变到 $n-1$ 得到的。第六个等式是从伯努利定理得出的。第七个等式是由于 $p+q=1$。由这个计算，可以断言 X 的期望值等于 np。这意味着在 n 次伯努利实验中预期的成功次数是 np。 ■

定理 2 建立了期望值的某些有用的性质，包括随机变量之和的期望值就是它们的期望值之和。

定理 2 如果 X 和 Y 是在样板空间 S 上的随机变量，那么 $E(X+Y)=E(X)+E(Y)$。此外，如果 X_i 是 S 上的随机变量，$i=1,2,\cdots,n$，n 是正整数，那么 $E(X)=E(X_1)+E(X_2)+\cdots+E(X_n)$。进而如果 a 和 b 是实数，那么 $E(aX+b)=aE(X)+b$。

证 第一个结果可直接从期望值的定义得出，因为

$$E(X+Y)=\sum_{s\in S}p(s)(X(s)+Y(s))$$

$$= \sum_{s \in S} p(s)X(s) + \sum_{s \in S} p(s)Y(s)$$
$$= E(X) + E(Y)$$

从两个随机变量的情况出发使用数学归纳法就能容易地得到具有 n 个随机变量的结论。最后，由于$\sum_{s \in S} p(s) = 1$，有$E(aX + b) = \sum_{s \in S} p(s)(aX(s) + b) = a\sum_{s \in S} p(s)X(s) + b\sum_{s \in S} p(s) = aE(X) + b$。■

定理 2 对于计算期望值可能是很有用的，因为许多随机变量是简单随机变量的和，正如例 15 和 16 所显示的。

例 15　用定理 2 找出掷一对均匀的骰子时所出现的点数之和的期望值（在例 13 中没有使用这个定理也求出了这个值）。

解　设 X_1 和 X_2 是随机变量，其中$X_1((i,j)) = i$，$X_2((i,j)) = j$，使得 X_1 是第一个骰子上出现的点数，X_2 是第二个骰子上出现的点数。容易看出，因为$(1+2+3+4+5+6)/6 = 21/6 = 7/2$，所以$E(X_1) = E(X_2) = 7/2$。当掷两个骰子时出现的两个点数之和就是和 $X_1 + X_2$。根据定理 2，这个和的期望值是$E(X_1 + X_2) = E(X_1) + E(X_2) = 7/2 + 7/2 = 7$。■

例 16　在例 14 中，当执行 n 次伯努利实验时成功次数的期望值由直接计算证明是 np，这里 p 是每次实验成功的概率。证明这个结果也可以使用定理 2 得到。

解　设 X_i 是随机变量。如果 t_i 是成功，$X_i(t_1, t_2, \cdots, t_n) = 1$；如果 t_i 是失败，$X_i(t_1, t_2, \cdots, t_n) = 0$。$X_i$ 的期望值是$E(X_i) = 1 \cdot p + 0 \cdot (1-p) = p$，$i = 1, 2, \cdots, n$。令 $X = X_1 + X_2 + \cdots + X_n$ 使得 X 计数当执行 n 次伯努利实验时成功的次数。把定理 2 用于 n 个随机变量的和，就证明了$E(X) = E(X_1) + E(X_2) + \cdots + E(X_n) = np$。■

我们已经讨论了独立事件。我们现在将定义什么是两个随机变量的独立性。

4.5.9　独立随机变量

定义 6　随机变量 X 和 Y 在样本空间 S 上是独立的，如果

$$p(X(s) = r_1 \text{ 且 } Y(s) = r_2) = p(X(s) = r_1) \cdot p(Y(s) = r_2)$$

换句话说，对一切实数 r_1 和 r_2，$X(s) = r_1$ 且 $Y(s) = r_2$ 的概率等于$X(s) = r_1$ 的概率与 $Y(s) = r_2$ 的概率之积。

例 17　例 15 的随机变量 X_1 和 X_2 是独立的吗？

解　设$S = \{1,2,3,4,5,6\}$，i, j 属于 S。由于掷一对骰子有 36 个可能的结果并且每个结果是等可能的，故

$$p(X_1 = i \text{ 且 } X_2 = j) = 1/36$$

又由于第一个骰子出现 i 和第二个骰子出现 j 的概率都是 1/6，即 $p(X_1 = i) = 1/6$ 且 $p(X_2 = j) = 1/6$，从而有

$$p(X_1 = i \text{ 且 } (X_2 = j) = 1/36 = (1/6)(1/6) = p(X_1 = i)p(X_2 = j)$$

因此 X_1 和 X_2 是独立的。 ■

例 18 证明随机变量 X_1 和 $X = X_1 + X_2$ 不是独立的，其中 X_1 和 X_2 的定义在例 15 中给出。

解 因为 $X_1 = 1$ 的含义是第一个骰子出现点数为 1，这就推出两个骰子的点数之和不可能等于 12，所以 $p(X_1 = 1 且 X = 12) = 0$。另一方面，$p(X_1 = 1) = 1/6$ 和 $p(X = 12) = 1/36$。因此，$p(X_1 = 1 且 X = 12) \neq p(X_1 = 1) \cdot p(X = 12)$。这个反例证明了 X_1 和 X 不是独立的。 ■

两个独立的随机变量之积的期望值是它们的期望值之积，正如定理 3 所述。

定理 3 如果 X 和 Y 是样本空间 S 上的独立的随机变量，那么 $E(XY) = E(X)E(Y)$。

证 因为 X 和 Y 是独立的随机变量，由随机变量的定义得到

$$\begin{aligned}E(XY) &= \sum_{s\in S} X(s)Y(s)p(s)\\&= \sum_{r_1\in X(S), r_2\in Y(S)} r_1 r_2 \cdot p(X(s) = r_1 \text{ 且 } Y(s) = r_2)\\&= \sum_{r_1\in X(S), r_2\in Y(S)} r_1 r_2 \cdot p(X(s) = r_1) \cdot p(Y(s) = r_2)\\&= \left(\sum_{r_1\in X(S)} r_1 p(X(s) = r_1)\right)\cdot\left(\sum_{r_2\in Y(S)} r_2 p(Y(s) = r_2)\right)\\&= E(X)E(Y)\end{aligned}$$ □

定理得证。

4.5.10 方差

一个随机变量的期望值告诉我们的是它的平均值，但是并没有说明它的值的分布范围。例如，如果 X 和 Y 是集合 $S = \{1,2,3,4,5,6\}$ 上的随机变量，对所有的 $s \in S$ 有 $X(s) = 0$，且若 $s \in \{1,2,3\}$ 则 $Y(s) = -1$；若 $S \in \{4,5,6\}$ 则 $Y(s) = 1$ 那么 X 和 Y 的期望值都是 0。但是随机变量 X 永远等于 0，而随机变量 Y 总是与 0 相差 1。一个随机变量的方差帮助我们刻划一个随机变量的值的分布范围。

定义 7 设 X 是样本空间上的随机变量。X 的方差记为 $V(X)$，且

$$V(X) = \sum_{s\in S}(X(s) - E(X))^2 p(s)$$

X 的标准差定义为 $\sqrt{V(X)}$，记作 $\sigma(X)$。

下面的定理提供了关于随机变量的方差的一个有用的简单表达式。

定理 4 如果 X 是样本空间 S 上的随机变量，那么 $V(X) = E(X^2) - E(X^2)$。

证 注意到

$$\begin{aligned}V(X) &= \sum_{s\in S}(X(s) - E(X))^2 p(s)\\&= \sum_{s\in S} X(s)^2 p(s) - 2E(X)\sum_{s\in S} X(s)p(s) + E(X)^2\sum_{s\in S} p(s)\\&= E(X^2) - 2E(X)E(X) + E(X)^2\\&= E(X^2) - E(X)^2\end{aligned}$$

在倒数第二步我们使用了 $\sum_{s\in S} p(s) = 1$ 这一事实。 ■

例 19　一个伯努利实验成功则$X(t)=1$，失败则$X(t)=0$。如果 p 是成功的概率，那么随机变量 X 的方差是什么？

解　因为 X 取值只能为 0 和 1，因此$X^2(t)=X(t)$。于是，

$$V(X)=E(X^2)-E(X)^2=p-p^2=p(1-p)=pq$$ ■

例 20　随机变量$X((i,j))=2i$ 的方差是什么？这里的 i 和 j 分别是掷两个骰子时第一个骰子和第二个骰子上出现的点数。

解　我们将使用定理 4 求出 X 的方差。为此，我们需要找到 X 和X^2 的期望值。注意到当 $k=2$，4，6，8，10，12 时$p(X=k)$是 1/6，否则为 0，因而有

$$E(X)=(2+4+6+8+10+12)/6=7$$

和

$$E(X^2)=(2^2+4^2+6^2+8^2+10^2+12^2)/6=182/3$$

由定理 4 得

$$V(X)=E(X^2)-E(X)^2=182/3-49=35/3$$ ■

另一个有用的关于方差的事实是，两个独立的随机变量的和的方差是它们的方差之和。例如，这个结果可用于计算 n 个独立的伯努利实验结果的方差。

定理 5　如果 X 和 Y 是样本空间 S 上两个独立的随机变量，那么$V(X+Y)=V(X)+V(Y)$。此外，如果对于正整数 n，X_i 是 S 上两两独立的随机变量，$i=1,2,\cdots,n$，那么$V(X_1+X_2+\cdots+X_n)=V(X_1)+V(X_2)+\cdots+V(X_n)$。

证　由定理 4 有

$$V(X+Y)=E((X+Y)^2)-E(X+Y)^2$$

从而有

$$\begin{aligned}V(X+Y)&=E(X^2+2XY+Y^2)-(E(X)+E(Y))^2\\&=E(X^2)+2E((XY)+E(Y^2)-E(X)^2-2E(X)E(Y)-E(Y)^2\end{aligned}$$

因为 X 和 Y 是独立的，由定理 3 有$E(XY)=E(X)E(Y)$，从而得到

$$\begin{aligned}V(X+Y)&=(E(X^2)-E(X)^2+(E(Y^2)-E(Y)^2)\\&=V(X)+V(Y)\end{aligned}$$

具有 n 个两两独立的随机变量的情况可以使用数学归纳法证明；这个证明留给读者完成。 ■

例 21　设掷两个骰子时随机变量 X 的值是$X((i,j))=i+j$，其中 i 是第一个骰子出现的点数，j 是第二个骰子出现的点数。求 X 的方差和标准差。

解　设 X_1 和 X_2 是掷骰子的随机变量，其中$X_1((i,j))=i$，$X_2((i,j))=j$。那么正如例 17 所证明的，$X=X_1+X_2$ 和 X_1 与 X_2 都是独立的。由定理 5 得到$V(X)=V(X_1)+V(X_2)$。按例 20 类似的简单计算与本章后的补充练习 43 告诉我们，$V(X_1)=V(X_2)=35/12$。因此，$V(X)=35/12+35/12=35/6$ 且 $\sigma(X)=\sqrt{35/6}$。 ■

我们现在求随机变量的方差，该随机变量计数执行 n 次独立的伯努利实验时的成功次数。

例 22 当执行 n 次独立的伯努利实验时，什么是计数成功次数的随机变量的方差？这里 p 是每次实验成功的概率。

解 设 X_i 是随机变量，且若是 t_i 成功则$X_i((t_1,t_2,\cdots,t_n))=1$,若是 t_i 失败则 $X_i((t_1,t_2,\cdots,t_n))=0$。令 $X=X_1+X_2+\cdots+X_n$,那么 X 计数在 n 次实验中的成功次数。由定理 5 得到 $V(X)=V(X_1)+V(X_2)+\cdots+V(X_n)$。使用例 19，有 $V(X_i)=pq$, $i=1,2,\cdots,n$。从而得到 $V(X)=npq$。■

4.5.11 切比雪夫不等式

一个随机变量的取值与它的期望值距离多远？下面的定理叫做切比雪夫[⊖]不等式，它对随机变量的值与它的期望值之差超过某个指定量的概率提供了一个上界，有助于回答这个问题。

定理 6 *切比雪夫不等式* 设 X 是在样本空间 S 上的概率函数为 p 的随机变量。如果 r 是一个正实数，那么

$$p(|X(s)-E(X)|\geqslant r)\leqslant V(X)/r^2$$

证 设 A 是事件

$$A=\{s\in S\,|\,|X(s)-E(X)|\geqslant r\}$$

我们想要证明的是$p(A)\leqslant V(X)/r^2$ 。注意

$$\begin{aligned}V(X)&=\sum_{s\in S}(X(s)-E(X))^2p(s)\\&=\sum_{s\in A}(X(s)-E(X))^2p(s)+\sum_{s\notin A}(X(s)-E(X))^2p(s)\end{aligned}$$

在这个表达式中的第二个和是非负的，因为它的每个被加数是非负的。又因为对于 A 中的每个元素 s，有$(X(s)-E(X))^2\geqslant r^2$,这个表达式的第一个和至少是$\sum_{s\in A}r^2p(s)$。因此，$V(x)\geqslant\sum_{s\in A}r^2p(s)=r^2p(A)$。这正是我们想证明的。□

尽管切比雪夫不等式能用于任何的随机变量，但对于随机变量的值大大超过其均值的概率往往不能提供一个实际的估计。这可以由下面的例子说明。

⊖ 切比雪夫（Pafnuty Lvovich Chebyshev，1821—1894） 切比雪夫诞生在俄罗斯。他的父亲是一个曾经与拿破仑战斗过的退役军官。1832 年，这个家庭带着九个孩子搬到莫斯科。他在莫斯科的家里学完了高中课程，进入了莫斯科大学的物理学数学系。作为一个学生，他提出了一种新的方法求方程的近似根。1841 年他从莫斯科大学毕业，获得数学学位，并且继续他的学习，在 1843 年通过硕士考试并在 1846 年完成他的硕士论文。

1847 年，切比雪夫受聘为圣彼得堡大学的一个助教。在 1847 年他写了一篇论文并通过答辩。1860 年他成为圣彼得堡大学的教授，并一直工作到 1882 年。他在 1849 年写的有关同余理论的著作对数论的发展是很有影响的。他关于素数分布的研究工作是开创性的。他证明了贝川（Bertrand）的猜想，即对每个整数 $n>3$，存在一个在 n 和 $2n-2$ 之间的素数。切比雪夫提出了一些新的思想，后来用这些思想证明了素数定理。切比雪夫用多项式作函数逼近的工作，广泛地用于计算机中对函数的求值。切比雪夫也对力学感兴趣。他研究了怎样通过机械耦合将旋转运动转换成直线运动。切比雪夫用三个连杆的近似直线运动实现平行运动。

例 23 设 X 是当掷一个均匀骰子时的随机变量，X 的值就是出现的数字。我们有 $E(X)=7/2$（见例 15）和 $V(X)=35/12$（见例 20）。因为 X 的可能取值是 1,2,3,4,5 和 6，$E(X)=7/2$，X 不可能比它的均值多 5/2。因此，如果 $r>5/2, p(|X-7/2|\geqslant r)=0$。由切比雪夫不等式知道 $p(|X-7/2|\geqslant r)\leqslant(35/12)/r^2$。例如，当 $r=3$ 时，切比雪夫不等式告诉我们 $p(|X-7/2|\geqslant 3)\leqslant(35/12)/9=35/108$。这是一个很差的估计，因为 $p(X-7/2|\geqslant 7/2)=0$。 ■

4.5.12 平均状态下的计算复杂性

计算一个算法在平均状态下的计算复杂性，可以转变成计算一个随机变量的期望值。设一个实验的样本空间是可能输入 $a_j(j=1,2,\cdots,n)$ 的集合，且令随机变量 X 对 a_j 赋值是 a_j 作为输入时该算法用到的操作次数。基于我们对输入的了解，对每个可能的输入 a_j 赋给一个概率 $p(a_j)$。那么该算法在平均状态下的复杂性是

$$E(X)=\sum_{j=1}^{n}p(a_j)X(a_j)$$

这就是 X 的期望值。

在例 24 中我们将显示怎样求线性搜索算法在不同假设下的计算复杂性，这些假设与被搜索的元素在表中的概率相关。

例 24 *线性搜索算法在平均状态下的计算复杂性* 给定元素 x 和 n 个不同实数的表。在 2.1 节描述的线性搜索算法通过把这个元素与表中的每个元素进行比较来查找 x。当 x 被找到或者检查了所有的元素并确定 x 不在表中时算法结束。如果 x 在表中的概率是 p，并且 x 是表中 n 个元素的概率相等，那么这个线性搜索算法在平均状态下的复杂度是什么？（存在 $n+1$ 种可能的输入：这个数在表中有 n 种，不在表中也作为一种。）

解 在 2.2 节例 4 我们证明了如果 x 等于表中的第 i 个元素要用 $2i+1$ 次比较，在 2.2 节例 2 中又证明了如果 x 不在表中要用 $2n+2$ 次比较。x 等于表中第 i 个元素 a_i 的概率是 p/n，x 不在表中的概率是 $q=1-p$。从而得到线性搜索算法在平均状态下的计算复杂性是

$$\begin{aligned}E&=3p/n+5p/n+\cdots+(2n+1)p/n+(2n+2)q\\&=\frac{p}{n}(3+5+\cdots+(2n+1))+(2n+2)q\\&=\frac{p}{n}((n+1)^2-1)+(2n+2)q\\&=p(n+2)+(2n+2)q\end{aligned}$$

（第三个等式是由于 3.2 节的例 2。）例如，当 x 保证在表中时，有 $p=1$（对每个 i，$x=a_i$ 的概率是 $1/n$）和 $q=0$。因此 $E=n+2$，正如我们在 2.2 节例 4 所证明的。

当 x 在表中的概率 p 是 1/2 时，可知 $q=1-p=1/2$，从而 $E=(n+2)/2+n+1=(3n+4)/2$。类似地，如果 x 在表中的概率是 3/4，我们有 $p=3/4$ 和 $q=1/4$，因此 $E=3(n+2)/4+(n+1)/2=(5n+8)/4$。

最后，当 x 保证不在表中时，有 $p=0$ 和 $q=1$。从而得到 $E=2n+2$，这并不奇怪，因为我们必须搜索整个的表。 ■

练习

1. 当掷一个不均匀的硬币时如果出现头像的可能性是不出现头像可能性的 3 倍，那么出现头像的概率是多少？不出现头像的概率是多少？
2. 当掷一个不均匀的骰子时如果出现 3 点的可能性是其他 5 点数中每个点数的 2 倍，求出每种结果的概率。
3. 当掷一个不均匀的骰子时如果出现 2 或 4 点的可能性是出现其他 4 个点数中某个数的 3 倍，求出每种结果的概率。
4. 证明当结果是等可能的时候，条件(i)和(ii)在拉普拉斯的概率定义下是满足的。
5. 掷一对骰子，第一个骰子出现 4 点的概率是 2/7，第二个骰子出现 3 点的概率是 2/7，且每个骰子出现其他点数的概率是 1/7。当 2 个骰子被掷时点数之和等于 7 的概率是多少？
6. 假设 E 和 F 是事件，满足 $p(E)=0.8, p(F)=0.6$。证明 $p(E\cap F)\geqslant 0.4$。
7. 证明如果 E 和 F 是事件，那么 $p(E\cap F)\geqslant p(E)+p(F)-1$。这就是邦弗罗尼不等式。
8. 使用数学归纳法证明下述一般性的邦弗罗尼不等式：

$$p(E_1\cap E_2\cap\cdots\cap E_n)\geqslant p(E_1)+p(E_2)\cdots+p(E_n)-(n-1)$$

   其中 $E_1,E_2,\cdots,E_n$ 是 n 个事件。
9. 证明如果 $E_1,E_2,\cdots,E_n$ 是一个有限样本空间的事件，那么

$$p(E_1\cup E_2\cup\cdots\cup E_n)\leqslant p(E_1)+p(E_2)+\cdots+p(E_n)$$

   这就是布尔不等式。
10. 如果 E 和 F 是独立的事件，证明 $\overline{E}$ 和 $\overline{F}$ 也是独立的事件。
11. 如果 E 和 F 是独立的事件，证明或反证 $\overline{E}$ 和 F 也必须是独立的事件。



练习 12～14 涉及了一组人里至少两个人有相同生日的概率。

12. 假定所有的 366 天作为生日具有相同的可能性，2 个人有相同生日的概率是多少？

*13. a) 假定 366 天的每一天作为生日是等可能的，那么在一组 n 个人里至少 2 个人有相同生日的概率是多少？[提示：找出在一组 n 个人里所有人的生日都不相同的概率。]
   b) 要使得至少两个人的生日相等的概率大于 1/2 需要多少个人？

*14. 只有闰年有 2 月 29 日。能被 4 整除但不能被 100 整除的都是闰年，能被 100 整除但不被 400 整除的不是闰年，能被 400 整除的是闰年。
   a) 对于生日应该用哪种概率分布来反映 2 月 29 日出现次数的多少？
   b) 用这个概率分布回答练习 13 的 a) 中所问的问题。

15. 给定掷硬币第一次的头像在上，当一个均匀的硬币被掷 5 次时恰好 4 次头像在上的条件概率是什么？
16. 给定掷硬币第一次的头像在下，当一个均匀的硬币被掷 5 次时恰好 4 次头像在上的条件概率是什么？
17. 给定第一位是 1，随机产生 4 位二进制串并使得它至少包含 2 个连续的 0 的条件概率是什么？

18. 随机产生3位二进制串，设 E 是这个串含有奇数个1的事件，F 是这个串以1开始的事件。E 和 F 是独立的吗？
19. 设 E 和 F 分别表示有 n 个孩子的家庭同时有男孩和女孩以及至多有1个男孩的事件。在下述每种条件下 E 和 F 是独立的吗？
    a) $n=2$　　b) $n=4$　　c) $n=5$
20. 假定一个孩子是男孩的概率是0.51，且诞生在一个家庭的孩子的性别是独立的。一个家庭有5个孩子，那么
    a) 恰有3个男孩的概率是什么？
    b) 至少有1个男孩的概率是什么？
    c) 至少有1个女孩的概率是什么？
    d) 所有的孩子有相同性别的概率是什么？
21. 一组6个人玩“单人出局”的游戏来确定谁买茶点。每个人掷一个均匀的硬币。如果一个人掷出的结果不和组中任何其他人相同，这个人就必须买茶点。在掷过一次硬币以后出现这种单人出局的概率是多少？
22. 随机产生不包含0的10位的二进制串，如果每位的产生是独立的。求出下列每种情况下的概率：
    a) 一位为0和为1是等可能的。
    b) 一个为1的概率是0.6。
    c) 第 i 位为1的概率是 $1/2^i$，$i=1, 2, 3, \cdots, 10$。
23. 求有5个孩子的家庭没有男孩的概率，如果孩子的性别是独立的，且
    a) 一个男孩和一个女孩是等可能的。
    b) 一个男孩的概率是0.51。
    c) 第 i 个孩子是男孩的概率是 $0.51-(i/100)$。
24. 随机产生一个以1开始或以00结尾的10位二进制串，如果每位的产生是独立的，分别求在22题a),b)和c)同样条件下的概率。
25. 按照习题23a),b)和c)同样的条件，分别求出有5个孩子的家庭中第1个孩子是男孩或者最后2个孩子是女孩的概率。
26. 求出在下述每种情况下执行 n 次独立的伯努利实验时的概率，其中每次实验的成功概率为 p。
    a) 没有1次成功的概率。
    b) 至少1次成功的概率。
    c) 至多1次成功的概率。
    d) 至少2次成功的概率。
27. 求出在下述每种情况下执行 n 次独立的伯努利实验时的概率，其中每次实验的成功概率为 p。
    a) 没有1次失败的概率。
    b) 至少1次失败的概率。
    c) 至多1次失败的概率。
    d) 至少2次失败的概率。

28. 当一个均匀的硬币被掷 10 次时，预期出现多少次头像在上？
29. 当一个均匀的骰子被掷 10 次时，预期出现多少次 6 点？
30. 一个硬币是不均匀的，使得掷出头像在上的概率是 0.6，当掷 10 次时预期出现多少次头像在上？
31. 掷 2 个不均匀的骰子，其中 3 点出现的次数是其他每个点数的 2 倍。2 个骰子预期出现的点数和是什么？
32. 如果彩票包含了从集合{1,2…,50}选出的 6 个中奖数字就赢奖 1000 万美元，否则不中奖，那么买 1 美元彩票中奖的期望值是多少？
33. 离散数学课程的期末考试有 50 道真假判断题，每道题 2 分；还有 25 道多选题，每道题 4 分。琳达正确回答判断题的概率是 0.9，正确回答多选题的概率是 0.8。她在期末考试预期的分数是多少？
34. 当掷 3 个均匀的骰子时预期出现的数字和是多少？
35. 有 n 个不同整数的表，假设 x 在这个表中的概率是 2/3，且 x 等于表中任何元素的概率相等。求出线性搜索算法找 x 或确定它不在表中所用的平均比较次数。
*36. 有 n 个不同整数的表，假设 x 是表中第 i 个元素的概率为 $i/[n(n+1)]$。求由线性搜索算法找 x 或确定 x 不在表中使用的平均比较次数。

本节我们已经研究了具有有限多个结果的实验。在练习 37～40 我们将研究具有可数多个结果的实验。掷一个硬币直到它不出现头像为止。这个实验的样本空间是

$$\{T, HT, HHT, HHHT, HHHHT, \cdots\}$$

硬币不出现头像的概率是 p。

37. 在掷 n 次硬币后实验结束，即出现 $n-1$ 次头像和 1 次非头像的概率是多少？
38. 证明掷硬币所有可能结果的概率之和是 1。
39. 至多掷 n 次硬币就使实验结束的概率是多少？
40. 使实验结束所需要掷硬币次数的期望值是多少？
41. 在一个晚会上，n 个人把他们的帽子挂在柜子的帽架上。帽子是乱放的，且每个人随机地选择一顶。预期有多少人能够恰好选到自己的帽子？
42. 设 $X(s)$ 是随机变量，对所有的 $s \in S$，$X(s)$ 是非负整数，且 A_k 是满足 $X(s) \geqslant k$ 的事件。证明 $E(X)=\sum_{k=1}^{\infty} p(A_k)$。
43. 当一个均匀的硬币被掷 10 次时出现头像在上次数的方差是什么？
44. 当一个均匀的骰子被掷 10 次时出现 6 点的次数的方差是什么？
45. 设 X_n 是掷 n 个硬币时计数非头像次数和头像次数之差的随机变量。
    a) X_n 的期望值是什么？
    b) X_n 的方差是什么？
46. 提供一个例子说明当两个随机变量不独立时，它们的和的方差不一定等于它们的方差之和。
47. 设 X 是样本空间 S 上的一个随机变量，且对所有的 $s \in S$ 有 $X(s) \geqslant 0$。证明对每个正实数 a 有 $p(X(s) \geqslant a) \leqslant E(X)/a$。这个不等式叫马尔可夫不等式。
48. 假设一个灌装厂一天灌装苏打饮料的听数是一个随机变量。它的期望值是 10000，方差

是 1000。

a) 使用马尔可夫不等式（练习 47）得到该厂在某一天灌装听数超过 11000 的概率的上界。

b) 使用切比雪夫不等式得到该厂在某一天灌装听数在 9000 到 11000 之间的概率的下界。

49. 假设一个回收中心一天回收的罐头盒数是一个随机变量，它的期望值是 50000，方差是 2500。
    a) 使用马尔可夫不等式（练习 47）得到该中心在某一天回收罐头盒超过 55000 的概率的上界。
    b) 使用切比雪夫不等式提供该中心在某一天回收的罐头盒数在 40000 到 60000 之间的概率的下界。

两个随机变量 X 和 Y 在样本空间 S 的协方差记作 $\mathrm{Cov}(X,Y)$，定义为随机变量 $(X-E(X))(Y-E(Y))$ 的期望值，即 $\mathrm{Cov}(X,Y)=E((X-E(X))(Y-E(Y)))$。

50. 证明 $\mathrm{Cov}(X,Y)=E(XY)-E(X)E(Y)$，并使用这一结果证明，如果 X 和 Y 是独立的随机变量则 $\mathrm{Cov}(X,Y)=0$。
51. 证明 $V(X+Y)=V(X)+V(Y)+2\mathrm{Cov}(X,Y)$。
52. 如果 X 和 Y 是具有 $X((i,j))=2i$ 和 $Y((i,j))=i+j$ 的随机变量，其中 i 和 j 是掷两个均匀的骰子时出现在第一和第二个骰子上的点数，求 $\mathrm{Cov}(X+Y)$。

4.6 一般性的排列和组合

4.6.1 引言

在许多计数问题里元素可以被重复使用。例如，一个字母或一个数字可以在一个车牌中多次使用。当选一打多纳圈时，每种可以被重复地选择。这与本章前面讨论的计数问题形成对照，因为那里我们只考虑每个项至多可以使用一次的排列和组合。在这一节我们将显示怎样求解元素可以多次使用的计数问题。

还有，某些计数问题涉及到不可区别的元素。例如，为计数单词 SUCCESS 的字母可能被重新排列的方式数，必须考虑相同字母的放置。这又与前面讨论的所有元素都被认为是不同的计数问题大相径庭。在这一节，我们将描述怎样求解某些元素是不可区别的计数问题。

此外，在这一节我们也将解释怎样求解另一类重要的计数问题，即计数把不同的元素放人盒子的方法数的问题。这种问题的一个例子是把扑克牌发给 4 个玩牌人的不同的方式数。

把在本章前面描述的方法与这一节引人的方法一起考虑，就构成一个求解广泛的计数问题的有用的工具箱。当把第 5 章讨论的新的方法再加到这个库里，你将能够求解在广泛的研究领域中产生的大多数计数问题。

4.6.2 有重复的排列

当允许重复时考虑下面计数问题的例子。

例 1 用英语字母可以构成多少个 n 位字符串？

解 因为有26个字母，且每个字母可以被重复使用，由乘积法则可以看出存在 26^n 个 n 位字符串。 ■

下面的问题涉及概率，也涉及有重复的排列。

例2 一个缸包含5个红球和7个蓝球。如果一个球取出以后又放回缸里，那么从这个缸里连续取出3个红球的概率是多少？

解 因为每次取球时有5个红球在缸里，根据乘积法则，成功的结果数（即取出3个红球的方式数）是 5^3。又因为每次取球时缸里都是12个球，取球的结果总数是 12^3。于是，所求的概率是 $5^3/12^3=125/1728$。这是一个放回抽样的例子。 ■

下面的定理给出了当允许重复时一个 n 元素集合的 r-排列数。

定理1 具有 n 个物体的集合允许重复的 r-排列数是 n^r。

证 当允许重复时，在 r-排列中对 r 个位置中的每个位置有 n 种方式选择集合的元素，因为对每个选择，所有 n 个物体都是有效的。因此，由乘积法则，当允许重复时存在 n^r 个 r-排列。

4.6.3 有重复的组合

考虑下面元素允许重复的组合实例。

例3 从包含苹果、橙子和梨的碗里选4个水果。如果选择水果的顺序无关，且只关心水果的类型而不管是该类型的哪一个水果，那么当碗中每类水果至少有4个时有多少种选法？

解 为了求解这个问题，我们列出选择水果的所有可能的方式。共有15种方式：

4个苹果	4个橙子	4个梨
3个苹果，1个橙子	3个苹果，1个梨	3个橙子，1个苹果
3个橙子，1个梨	3个梨，1个苹果	3个梨，1个橙子
2个苹果，2个橙子	2个苹果，2个梨	2个橙子，2个梨
2个苹果，1个橙子，1个梨	2个橙子，1个苹果，1个梨	2个梨，1个苹果，1个橙子

这个解是从3个元素的集合{苹果,橙子,梨}中允许重复的4-组合数。 ■

为求解这种类型的更复杂的计数问题，我们需要计数一个 n 元素集合的 r-组合的一般方法。在例4中，我们将给出这一方法。

例4 从包含1美元、2美元、5美元、10美元、20美元、50美元及100美元的钱袋中选5张纸币，有多少种方式？假定不管纸币被选的次序，同种币值的纸币都是不加区别的，并且至少每种纸币有5张。

解 因为纸币被选的次序是无关的且7种不同类型的纸币都可以选5次，问题涉及的是计数从7个元素的集合中允许重复的5-组合数。列出所有的可能性将是乏味的，因为存在许多的解。相反，我们将给出一种方法来计数允许重复的组合数。

假设一个零钱盒子有7个隔间，每个保存一种纸币，如图4-6所示。这些隔间被6块隔

板分开，正如图中所画的。每选择 1 张纸币就对应于在相应的隔间里放置 1 个标记。图 4-7 针对选择 5 张纸币的 3 种不同方式给出了这种对应，其中的竖线表示 6 个隔板，星表示 5 张纸币。

选择 5 张纸币的方法数对应于安排 6 条竖线和 5 颗星的方法数。因此，选择 5 张纸币的方法数就是从 11 个可能的位置选 5 颗星位置的方法数。这对应于从含 11 个物体的集合中无序地选择 5 个物体的方法数，可以有 $C(11,5)$ 种方式。因此存在



图 4-6　有 7 种类型纸币的零钱盒

$$C(11,5)=\frac{11!}{5!\ 6!}=462$$

种方式从有 7 类纸币的袋中选择 5 张纸币。 ■

下面的定理将这个讨论一般化。

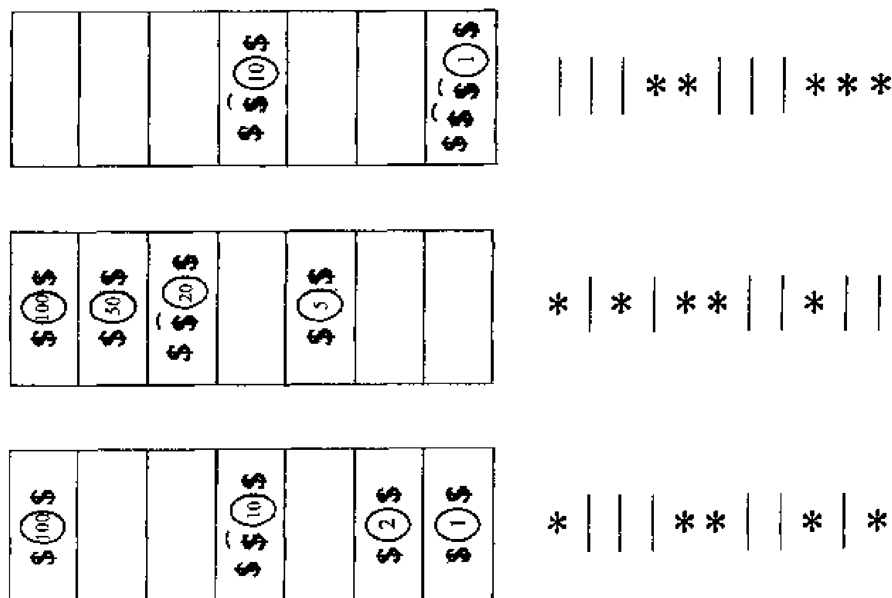


图 4-7　选择 5 张纸币方式的实例

定理 2　从 n 个元素的集合中允许重复的 r-组合有 $C(n+r-1,r)$ 个。

证　当允许重复时 n 元素集合的每个 r-组合可以用 $n-1$ 条竖线和 r 颗星的表表示。这 $n-1$ 条竖线是用来标记 n 个不同的单元。每当集合的第 i 个元素出现在组合中，第 i 个单元就包含一颗星。例如，4 元素集合的一个 6-组合用 3 条竖线和 6 颗星来表示。这里

* * | * | | * * *

代表了恰包含 2 个第一元素、1 个第二元素、0 个第三元素和 3 个第四元素的组合。

正如我们已经看到的，包含 $n-1$ 条竖线和 r 颗星的每一个不同的表对应于 n 元素集合的允许重复的一个 r-组合。这种表的个数是 $C(n-1+r,r)$，因为每个表对应于从包含 r 颗星和 $n-1$ 条竖线的 $n-1+r$ 个位置中取 r 个位置来放 r 颗星的一种选择。 □

下面的例子说明定理 2 是怎样使用的。

例 5 设一家甜点店有四种不同类型的甜点，那么从中选 6 块甜点有多少种不同的方式？假定只关心甜点的类型，而不管是哪一块甜点或者选择的次序。

解 选择 6 块甜点的方式数是具有 4 类元素集合的 6-组合数。由定理 2，这等于

$$C(4+6-1,6)=C(9,6)$$

由于

$$C(9,6)=C(9,3)=\frac{9\cdot8\cdot7}{1\cdot2\cdot3}=84$$

选择 6 块甜点的不同方式数有 84 种。 ■

定理 2 也可以用于求给定线性方程的整数解的个数。这可以由下面的例子来说明。

例 6 方程 $x_1+x_2+x_3=11$ 有多少个解？其中 x_1，x_2 和 x_3 是非负整数。

解 为计数解的个数，注意到一个解对应了从 3 元素集合中选 11 个元素的方式，以使得 x_1 选自第一类，x_2 选自第二类，x_3 选自第三类。因此，解的个数等于 3 元素集合允许重复的 11-组合数。由定理 2 存在

$$C(3+11-1,11)=C(13,11)=C(13,2)=\frac{13\cdot12}{1\cdot2}=78$$

个解。

当对变元加上限制时也可以求出这个方程的解的个数。例如，当变元是满足 $x_1\geqslant1$，$x_2\geqslant2$，$x_3\geqslant3$ 的整数时，我们也可以求出这个方程的解的个数。满足此限制的方程的解对应于 11 个项的选择，使得项 x_1 取自第一类，项 X_2 取自第二类，项 x_3 取自第三类，并且第一类元素至少取 1 个，第二类元素至少取 2 个，第三类元素至少取 3 个。因此，先选 1 个第一类的元素，2 个第二类的元素，3 个第三类的元素；然后再多选 5 个元素。由定理 2，可以用

$$C(3+5-1,5)=C(7,5)=C(7,2)=\frac{7\cdot6}{1\cdot2}=21$$

种方式做到。于是，对给定限制的方程存在 21 个解。 ■

下面的例子说明了怎样计数在确定变量值时产生的允许重复的组合数，这个变量当每次执行某个嵌套循环时它的值都会增加。

例 7 在下面的伪码被执行后 k 的值是什么？

```
k := 0
for i_1 := 1 to n
  for i_2 := 1 to i_1
    ⋮
  for i_m := 1 to i_{m-1}
    k := k + 1
```

解 k 的初值是0，且对于一组满足

$$1\leqslant i_m\leqslant i_{m-1}\leqslant\cdots\leqslant i_1\leqslant n$$

的整数 i_1，i_2，…，i_m，每次执行这个嵌套循环时 k 的值就加1。这种整数的组数是从$\{1,2,\cdots,n\}$中允许重复地选择 m 个整数的方式数。(因为一旦这组整数选定以后，如果按非降序排列它们，这就唯一地确定了一组对 i_m，i_{m-1}，…，i_1 的赋值。相反，每个这样的赋值对应了一个唯一的无序集合。) 所以由定理2得出在代码被执行后$k=C(n+m-1,m)$。■

从一个 n 元素集合中，允许重复和不重复地选择 r 个元素，其有序和无序的选择数的公式在表4-1给出。

表4-1 允许和不允许重复的组合与排列

类 型	允许重复	公 式
r-排列	不	$\frac{n!}{(n-r)!}$
r-组合	不	$\frac{n!}{r!(n-r)!}$
r-排列	是	n^r
r-组合	是	$\frac{(n+r-1)!}{r!(n-1)!}$

4.6.4 具有不可区别物体的集合的排列

在计数问题中某些元素可能是没有区别的。在这种情况下必须小心避免重复计数。考虑下面的例子。

例8 重新排序单词SUCCESS中的字母能构成多少个不同的串？

解 因为SUCCESS中的某些字母是重复的，答案并不是7个字母的排列数。这个单词包含3个S，2个C，1个U和1个E。为确定重新排序单词中的字母能构成多少个不同的串，首先，注意到3个S可以用$C(7,3)$种不同的方式放在7个位置中，剩下4个空位。然后可以用$C(4,2)$种方式放2个C，留下2个空位。又可以用$C(2,1)$种方式放U，留下1个空位。因此，放E只有$C(1,1)$种方式。从而，由乘积法则，产生的不同的串数是

$$\begin{aligned}C(7,3)C(4,2)C(2,1)C(1,1)&=\frac{7!}{3!4!}\cdot\frac{4!}{2!2!}\cdot\frac{2!}{1!1!}\cdot\frac{1!}{1!0!}\\&=\frac{7!}{3!2!1!1!}\\&=420\end{aligned}$$

■

使用和前面例子同样的推理，能够证明下面的定理。

定理3 设类型1的相同的物体有 n_1 个，类型2的相同的物体有 n_2 个，…，类型 k 的相同的物体有 n_k 个，那么 n 个物体的不同排列数是

$$\frac{n!}{n_1!n_2!\cdots n_k!}$$

证 为确定排列数，首先注意到可以用$C(n,n_1)$种方式在 n 个位置中放类型1的 n_1 个

物体，剩下 $n-n_1$ 个空位。然后用 $C(n-n_1,n_2)$ 种方式放类型 2 的物体，剩下 $n-n_1-n_2$ 个空位。继续放类型 3 的物体，…，类型 $k-1$ 的物体，直到最后可用 $C(n-n_1-n_2-\cdots-n_{k-1},n_k)$ 种方式放类型 k 的物体。因此由乘积法则，不同排列的总数是

$$C(n,n_1)C(n-n_1,n_2)\cdots C(n-n_1-n_2-\cdots-n_{k-1},n_k)$$
$$=\frac{n!}{n_1!(n-n_1)!}\cdot\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!}\cdots\frac{(n-n_1-\cdots-n_{k-1})!}{n_k!0!}$$
$$=\frac{n!}{n_1!n_2!\cdots n_k!}$$ □

4.6.5 把物体放入盒子

有些计数问题可以通过枚举把不同的物体放入不同的盒子的方式数来求解。考虑下面的例子，其中物体是牌且"盒子"是玩牌人的手。

例 9 *有多少种方式把 52 张标准的扑克牌发给 4 个人使得每个人 5 张牌？*

解 我们将使用乘积法则求解这个问题。开始，第一个人得到 5 张牌可以有 $C(52,5)$ 种方式。第二个人得到 5 张牌可以有 $C(47,5)$ 种方式，因为只剩下 47 张牌。第三个人得到 5 张牌可以有 $C(42,5)$ 种方式。最后，第四个人得到 5 张牌可以有 $C(37,5)$ 种方式。因此，发给 4 个人每人 5 张牌的方式总数是

$$C(52,5)C(47,5)C(42,5)C(37,5)=\frac{52!}{47!5!}\cdot\frac{47!}{42!5!}\cdot\frac{42!}{37!5!}\cdot\frac{37!}{32!5!}$$
$$=\frac{52!}{5!5!5!5!32!}$$ ■

注意 例 9 的解等于 52 个物体的排列数，这些物体分成 5 个不同的类，其中四类，每类有 5 个相同的物体，第五类有 32 个物体。可以通过在这种排列和给人发牌之间定义一个一一对应来说明这个等式。为定义这个对应，首先把牌从 1 到 52 排序。然后将发给第一个人的牌与分配给第一类物体在排列中的位置对应。类似地，发给第二、第三和第四个人的牌分别与第二、第三、第四类的物体所分配的位置对应。没有发给任何人的牌与第五类物体所分配的位置对应。读者应该能够验证这是一个一一对应。

例 9 是涉及把不同的物体分配到不同的盒子的一个典型的问题。这些不同的物体是 52 张牌，5 个不同的盒子是 4 个人的手和其余的牌。涉及把不同的物体分配到不同的盒子的计数问题可以使用下面的定理求解。

定理 4 把 n 个不同的物体分配到 k 个不同的盒子使得 n_i 个物体放入盒子 $i(i=1,2,\cdots,k)$ 的方式数等于

$$\frac{n!}{n_1!n_2!\cdots n_k!}$$

定理的证明留给读者（见练习 43 和 44）。

练习

1. 从一个 3 元素集合中允许重复地有序选取 5 个元素有多少种不同的方式？

2. 从一个 5 元素集合中允许重复地有序选取 5 个元素有多少种不同的方式？
3. 6 个字母的字符串有多少个？
4. 每天一个学生从一堆包好的三明治中随机选 1 块三明治作为午饭。如果有 6 种三明治并且选择三明治的次序无关，在一周的 7 天里这个学生选择三明治有多少种不同的方式？
5. 分配 3 种工作给 5 个雇员，如果每个雇员可以得到 1 种以上的工作，那么有多少种不同的方式？
6. 从一个 3 元素集合中允许重复地无序选取 5 个元素有多少种不同方式？
7. 从一个 5 元素集合中允许重复地无序选取 3 个元素有多少种不同方式？
8. 从一个商店的 21 种多纳圈中选择 12 个多纳圈有多少种不同的方式？
9. 一个百吉饼店有洋葱百吉饼、罂粟子百吉饼、鸡蛋百吉饼、咸味百吉饼、粗制倮麦百吉饼、芝麻百吉饼、葡萄干百吉饼和普通百吉饼。有多少种方式选择
   a) 6 个百吉饼？
   b) 12 个百吉饼？
   c) 24 个百吉饼？
   d) 12 个百吉饼，并且每类至少有 1 个？
   e) 12 个百吉饼，并且至少有 3 个鸡蛋百吉饼和不超过 2 个咸味百吉饼？
10. 一个新月形面包店有普通新月形面包、樱桃新月形面包、巧克力新月形面包、杏仁新月形面包、苹果新月形面包和椰菜新月形面包。有多少种方式选择
    a) 12 个新月形面包？
    b) 36 个新月形面包？
    c) 24 个新月形面包，并且至少每类有 2 个？
    d) 24 个新月形面包，并且不超过 2 个椰菜的？
    e) 24 个新月形面包，并且至少 5 个巧克力的和至少 3 个杏仁的？
    f) 24 个新月形面包，并且至少 1 个普通的，至少 2 个樱桃的，至少 3 个巧克力的，至少 1 个杏仁的，至少 2 个苹果的和不超过 3 个椰菜的？
11. 一个小猪储钱罐包含 100 个相同的 1 美分和 80 个相同的 5 美分硬币，从中选 8 个硬币有多少种方式？
12. 如果一个小猪储钱罐有 1 美分、5 美分、10 美分、25 美分和 50 美分等硬币，那么 20 个硬币有多少种不同的组合？
13. 一个出版商有 3000 本离散数学书，如果这些书是没有区别的，那么将这些书存储在 3 个库房有多少种方式？
14. 设 x_1，x_2，x_3 和 x_4 是非负整数，方程

    $$x_1 + x_2 + x_3 + x_4 = 17$$

    有多少个解？
15. 方程

    $$x_1 + x_2 + x_3 + x_4 + x_5 = 21$$

    有多少个解？其中 $x_i (i = 1,2,3,4,5)$ 是非负整数，并且使得
    a) $x_1 \geqslant 1$
    b) $x_i \geqslant 2$，$i = 1, 2, 3, 4, 5$

c) $0 \leqslant x_1 \leqslant 10$

d) $0 \leqslant x_1 \leqslant 3$，$1 \leqslant x_2 < 4$，$x_3 \geqslant 15$

16. 方程

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 29$$

有多少个解？其中$x_i(i=1,2,3,4,5,6)$是非负整数，并且使得

a) $x_i \geqslant 1$，$i=1, 2, 3, 4, 5, 6$

b) $x_1 \geqslant 1$，$x_2 \geqslant 2$，$x_3 \geqslant 3$，$x_4 \geqslant 4$，$x_5 > 5$，$x_6 \geqslant 6$

c) $x_1 \leqslant 5$

d) $x_1 < 8$，$x_2 > 8$

17. 有多少 10 位三进制数字（0，1 或 2）串恰含有 2 个 0、3 个 1 和 5 个 2？
18. 有多少 20 位十进制数字串含有 2 个 0、4 个 1、3 个 2、1 个 3、2 个 4、3 个 5、2 个 7 和 3 个 9？
19. 假设一个大家庭有 14 个孩子，包括 2 组三胞胎，3 组双胞胎，以及 2 个单胞胎。这些孩子坐在一排椅子上，如果相同的三胞胎或双胞胎的孩子不能互相区分，那么有多少种方式？
20. 不等式

$$x_1 + x_2 + x_3 \leqslant 11$$

有多少个解？其中 x_1，x_2 和 x_3 是非负整数［提示：引入辅助变量 x_4 使得 $x_1 + x_2 + x_3 + x_4 = 11$。］

21. 有多少个小于 1 000 000 的正整数其数字之和等于 19？
22. 有多少个小于 1 000 000 的正整数恰好一个数字等于 9 且数字之和等于 13？
23. 一次离散数学的期终考试有 10 道题。如果总分数是 100 且每道题至少 5 分，那么有多少种方式来分配这些题的分数？
24. n 个物体有 r 种不同的类型，证明有$C(n+r-q_1-q_2-\cdots-q_r-1, n-q_1-q_2-\cdots-q_r)$种不同的无序选择，使得该选择至少有 q_1 个 1 型的物体，q_2 个 2 型物体，…，q_r 个 r 型物体。
25. 如果被传送的二进制串必须以 1 开始，必须有另外 3 位 1（使得传送的 1 共有 4 位），必须包含总共 12 位 0，必须每个 1 后面至少跟随 2 个 0，那么有多少个不同的二进制串？
26. 使用 *MISSISSIPPI* 中的所有字母可以构造多少个不同的串？
27. 使用 *ABRACADABRA* 中的所有字母可以构造多少个不同的串？
28. 使用 *AARDVARK* 中的所有字母且所有的 3 个 *A* 必须连续，那么可以构造多少个不同的串？
29. 使用 *ORONO* 中的某些或全部字母可以构造多少个不同的串？
30. 使用 *SEERESS* 中的字母可以构造多少个至少含 5 个字符的串？
31. 用 *EVERGREEN* 中的字母可以构造多少个至少含 7 个字符的串？
32. 使用 6 个 1 和 8 个 0 可以构造多少个不同的二进制串？
33. 一个学生有 3 个芒果、2 个番木瓜和 2 个猕猴桃。如果这个学生每天吃 1 个水果，并且只考虑水果的类型，那么有多少种不同的方式吃完这些水果？

34. 一个教授把40本数学期刊放入4个盒子，每盒10本，分配这些期刊有多少种方式？
   a) 如果每个盒子被编号使得它们是可区分的。
   b) 这些盒子是相同的，使得它们是不可区分的。
35. 有多少种不同的方式在 xyz 空间上从原点(0,0,0)到达(4,3,5)点？这个旅行的每一步是在 x 正方向移动一个单位，y 正方向移动一单位，或者 z 正方向移动一个单位。(x，y，z 负方向的移动是禁止的，即不允许回头。)
36. 有多少种不同的方式在 $xyzw$ 空间上从原点(0,0,0,0)到达(4,3,5,4)点？这个旅行的每一步是在 x，y，z 或 w 正方向移动一个单位。
37. 把一副标准的52张扑克牌发给5个人，每个人得到7张牌，有多少种方式？
38. 在打桥牌时，把一副标准的52张牌发给4个人，有多少种不同发牌的方式？
39. 当把一副标准的52张牌发给4个人时，若使得每个人有一手包含1个 A 的牌，这种概率是多少？
40. 12本书放在4个不同的书架上有多少种方式？
   a) 如果这些书是同一种书。
   b) 如果所有的书都不同，并且考虑这些书在书架上的位置。[提示：把这件事分成12个任务完成，放每本书是一个任务。先用1，2，3，4表示这些书架，用 b_i，$i=1$，2，…，12，表示书。把 b_i 放到1，2，3，4中某个数的右边。]
41. n 本书放在 k 个不同的书架上有多少种方式？
   a) 如果这些书是同一种书。
   b) 如果所有的书都不同，并且考虑这些书在书架上的位置。
42. 12本书在一个书架上排成一排。从中选5本书并且使得没有2本书相邻有多少种方式？[提示：将选的书用竖线表示，没有选的书用星号表示，计数含5条竖线和7颗星且没有2条竖线相邻的序列数。]

*43. 通过先把物体放入第一个盒子，然后把物体放入第二个盒子，…，的方法，使用乘积法则证明定理4。

*44. 通过下面的方法证明定理4。有 n 个物体，其中类型为 i 的相同的物体有 n_i 个，$i=1$，2，…，k。先把这 n 个物体的排列和把这些物体放到 k 个盒子且使得盒子 i 含有 n_i 个物体的分配之间建立一一对应，这里的 $i=1$，2，…，k，然后使用定理3。

*45. 在这个练习中我们将通过在两个集合之间建立一一对应来证明定理2。这两个集合分别是集合 $S=\{1,2,\cdots,n\}$ 的允许重复的 r-组合的集合和集合 $T=\{1,2,3,\cdots,n+r-1\}$ 的 r-组合的集合。
   a) 把 S 的允许重复的 r-组合中的元素排成一个递增序列 $x_1\leqslant x_2\leqslant\cdots\leqslant x_r$。证明对这个序列的第 k 项加上 $k-1$ 而构成的序列是严格递增的。断言这个序列由 T 的 r 个不同的元素构成。
   b) 证明在 a) 所描述的过程在 S 的允许重复的 r-组合的集合与 T 的 r-组合的集合之间定义了一一对应。[提示：通过把 T 的满足 $1\leqslant x_1<x_2<\cdots<x_r\leqslant n+r-1$ 的 r-组合 $\{x_1,x_2,\cdots,x_r\}$，与从第 k 个元素减去 $k-1$ 得到的 S 的允许重复的 r-组合相联系，证明这个对应是可逆的]。
   c) 断言存在着 $C(n+r-1,r)$ 个 n 元素集合的允许重复的 r-组合。

46. 有多少种方式把 5 个不同的物体放到 3 个相同的盒子里？
47. 有多少种方式把 5 个相同的物体放到 3 个相同的盒子里？
48. 在$(x_1+x_2+\cdots+x_m)^n$ 的展开式中把所有的同类项合并以后有多少个不同的项？
*49. 证明多项式定理：如果 n 是正整数，则

$$(x_1+x_2+\cdots x_m)^n = \sum_{n_1+n_2+\cdots n_m=n} C(n,n_1,n_2,\cdots,n_m)x_1^{n_1}x_2^{n_2}\cdots x_n^{n_m}$$

其中

$$C(n,n_1,n_2,\cdots,n_m)=\frac{n!}{n_1!n_2!\cdots n_m!}$$

是多项式系数。
50. 求 $(x+y+z)^4$ 的展开式。
51. 求 $(x+y+z)^{10}$中的 $x^3y^2z^5$ 的系数。
52. 在 $(x+y+z)^{100}$的展开式中有多少个项？

4.7　生成排列和组合

4.7.1　引言

本章前几节已经描述了各种类型的排列和组合的计数方法，但是有时候需要生成排列和组合，而不仅仅是计数。考虑下面三个问题。第一，假设一个销售商必须访问 6 个城市。应该按照什么顺序访问这些城市而使得总的旅行时间最少？确定最好顺序的一种方法就是确定 6! =720 种不同顺序的访问时间并且选择具有最小旅行时间的访问顺序。第二，假定 6 个数的集合中某些数的和是 100。找出这些数的一种方法就是生成所有 $2^6=64$ 个子集并且检查它们的元素和。第三，假设一个实验室有 95 个雇员，一个项目需要一组 12 人组成的有 25 种特定技能的雇员。(每个雇员可能有一种或多种技能。)找出这组雇员的一种方法就是找出所有的 12 个雇员的小组，然后检查他们是否有所需要的技能。这些例子都说明为了求解问题常常需要生成排列和组合。

4.7.2　生成排列

任何 n 元素集合可以与集合$\{1,2,3,\cdots,n\}$建立一一对应。我们可以如下列出任何 n 元素集合的所有排列：生成 n 个最小正整数的排列，然后用对应的元素替换这些整数。已经建立了许多不同的算法来生成这个集合的 $n!$ 个排列。我们将要描述的算法是以$\{1,2,\cdots,n\}$的排列集合的字典顺序为基础的。按照这个顺序，如果对于某个 k，$1\leqslant k\leqslant n$，$a_1=b_1$，$a_2=b_2$，…，$a_{k-1}=b_{k-1}$，$a_k<b_k$，那么排列 $a_1a_2\cdots a_n$ 在排列 $b_1b_2\cdots b_n$ 的前边。换句话说，如果在 n 个最小正整数集合的两个排列不等的第一位置，一个排列的数小于第二个排列的数，那么这个排列按照字典顺序排在第二个排列的前边。

例 1　集合$\{1,2,3,4,5\}$的排列 23415 在排列 23514 的前边，因为这两个排列的两位相同，但第一排列在第三位置中的数是 4，小于第二排列在第三位置中的数 5。类似地，排列

41532 在排列 52143 的前边。■

生成{1,2,…,n}的排列的算法基础是从一个给定排列 $a_1a_2\cdots a_n$ 按照字典顺序构造下一个排列的过程。我们将说明怎样做到这一点。首先假设 $a_{n-1}<a_n$，交换 a_{n-1}和 a_n 可得到一个更大的排列。没有其他的排列既大于原来的排列并且又小于这个通过交换 a_{n-1}与 a_n 得到的排列。例如，在 234156 后面的下一个最大的排列是 234165。另一方面，如果 $a_{n-1}>a_n$，那么由交换这个排列中的最后两项不可能得到一个更大的排列。看看排列中的最后 3 个整数，如果 $a_{n-2}<a_{n-1}$，那么可以重新安排这后 3 个数而得到下一个最大的排列。a_{n-1}和 a_n 中比较小的数大于 a_{n-2}。先把这个数放在位置 $n-2$，然后把剩下的那个数和 a_{n-2}按照递增的顺序放到最后的两个位置。例如，在 234165 后面的下一个最大的排列是 234516。

另一方面，如果 $a_{n-2}>a_{n-1}$（且 $a_{n-1}>a_n$），那么不可能由安排在这个排列的最后三项而得到更大的排列。基于这个观察，可以描述一个一般的方法，对于给定的排列 $a_1a_2\cdots a_n$ 依据字典顺序来生成下一个最大的排列。首先，找到整数 a_j 和 a_{j+1}使得 $a_j<a_{j+1}$且

$$a_{j+1}>a_{j+2}>\cdots>a_n$$

即在这个排列中的最后一对相邻的整数，使得这个对的第一个整数小于第二个整数。然后，把 a_{j+1}，a_{j+2}，…，a_n 中大于 a_j 的最小的整数放到第 j 个位置，再按照递增顺序从位置 $j+1$ 到 n 列出 a_j，a_{j+1}，a_{j+2}，…，a_n 中其余的整数，这就得到依照字典顺序的下一个最大的排列。容易看出，没有其他的排列大于排列 $a_1a_2\cdots a_n$ 而小于这个新生成的排列。（对这一事实的验证留给读者作为练习。）

例 2 在 362541 后面按照字典顺序的下一个最大排列是什么？

解 使得 $a_j<a_{j+1}$的最后一对整数 a_j 和 a_{j+1}是 $a_3=2$ 和 $a_4=5$。排列在 2 右边大于 2 的最小整数是 $a_5=4$。因此 4 被放在第三位置。然后整数 2，5 和 1 依递增顺序放到最后 3 个位置，即这个排列的最后 3 个位置是 125。于是，下一个排列是 364125。■

为生成整数 1，2，…，n 的 $n!$ 个排列，按照字典顺序由最小的排列，即 $123\cdots n$ 开始，连续施用 $n!-1$ 次生成下一个最大排列的过程，就得到 n 个最小的整数按字典顺序的所有排列。

例 3 按字典顺序生成整数 1，2，3 的排列。

解 从 123 开始，由交换 3 和 2 得到下一个排列 132。下一步，因为 3>2 和 1<3，排列在 132 中的 3 个整数，把 3 和 2 中较小的放到第一个位置，然后按递增顺序把 1 和 3 放到位置 2 和 3 而得到 213。跟着 213 的是 231，它是由交换 1 和 3 得到的，因为 1<3。下一个最大的排列把 3 放在第一位置，后面是 1 和 2 按递增顺序排列，即 312。最后，交换 1 和 2 得到最后一个排列 321。■

算法 1 显示了在给定排列不是最大的排列 $n\ n-1\ n-2\cdots 21$ 时，在它的后面按照字典顺序找到下一个最大排列的过程。

算法 1　按字典顺序生成下一个最大排列

```
procedure next permutation(a1a2…an:{1,2,…,n}的排列不等于 n n-1 … 21)
  j := n - 1
while aj > aj+1
  j := j - 1
{j 是使得 aj < aj+1 的最大下标}
k := n
while aj > ak
  k := k - 1
{ak 是在 aj 右边大于 aj 最小正整数}
交换 aj 和 ak
r := n
s := j + 1
while r > s
begin
  交换 ar 和 as
  r := r - 1
  s := s + 1
end
{这把在第 j 位后边的排序尾部按递增顺序置放}
```

4.7.3　生成组合

怎样可以生成一个有穷集的元素的所有组合呢？由于一个组合仅仅就是一个子集，我们可以利用在 $\{a_1,a_2,\cdots,a_n\}$ 和 n 位二进制串之间的对应。

如果 a_k 在子集中，对应的二进制串在位置 k 有一个 1；如果 a_k 不在子集中，对应的二进制串在位置 k 有一个 0。如果可以列出所有的 n 位二进制串，那么通过在子集和二进制串之间的对应就可以列出所有的子集。

一个 n 位二进制串也是一个在 0 到 2^n-1 之间的整数的二进制展开式。按照它们的二进制展开式，作为整数根据递增顺序可以列出这 2^n-1 个二进制串。为生成所有的 n 位二进制展开式，从具有 n 个 0 的二进制串 000…00 开始。然后，继续找下一个最大的展开式，直到得到 111…11 为止。在每一步找下一个最大的二进制展开式时先确定从右边起第一个不是 1 的位置。然后把这个位置右边的所有的 1 变成 0，并且将这第一个 0（从右边数）变成 1。

例 4　找出在 10 0010 0111 后面的下一个最大的二进制串。

解　这个串从右边数不是 1 的第 1 位是从右边起的第 4 位。把这一位变成 1 并且将它后面所有的位变成 0。这就生成了下一个最大的二进制串 10 0010 1000。■

生成在 $b_{n-1}b_{n-2}\cdots b_1b_0$ 后面的下一个最大的二进制串的过程给出在算法 2 中。

算法 2 生成下一个最大的二进制串

```
procedure next bit string(b_{n-1}b_{n-2}…b_1b_0: 不等于 11…11 的二进制串)
i := 0
while b_i = 1
begin
    b_i := 0
    i := i + 1
end
b_i := 1
```

下面将给出生成集合{1,2,3,…,n}的 r-组合的算法。一个 r-组合可以表示成一个序列，这个序列按照递增的顺序包含这个子集中的元素。使用在这些序列上的字典顺序可以列出这些 r-组合。在 $a_1a_2\cdots a_r$ 后面的下一个组合可以按下面的方法得到：首先，找到序列中使得 $a_i \neq n-r+i$ 的最后元素 a_i，然后用 a_i+1 代替 a_i 且对于 $j=i+1, i+2, \cdots, r$ 用 $a_i+j-i+1$ 代替 a_j。请读者证明这就按字典顺序生成了下一个最大的组合。下面的例子说明了这个过程。

例 5 找出集合{1,2,3,4,5,6}在{1,2,5,6}后面的下一个最大的 4-组合。

解 在具有 $a_1=1$，$a_2=2$，$a_3=5$，$a_4=6$ 的项中使得 $a_i \neq 6-4+i$ 的最后的项是 $a_2=2$。为得到下一个最大的 4-组合，把 a_2 加 1 得 $a_2=3$。然后，置 $a_3=3+1=4$ 和 $a_4=3+2=5$。从而下一个最大的 4-组合是 {1, 3, 4, 5}。 ■

算法 3 用伪码给出了这个过程。

算法 3 按字典顺序生成下一个 r-组合

```
procedure next r-combination({a_1,a_2,…,a_r}: {1, 2, …n} 满足 a_1<a_2<…<a_r 的
    不等于{n - r + 1,…,n}的真子集)
i := r
while a_i = n - r + i
    i := i - 1
a_i := a_i + 1
for j = i + 1 to r
    a_j := a_i + j - i
```

练习

1. 找出按照字典顺序跟在下面每一个排列后面的下一个最大的排列。

   a) 1432　　b) 54123　　c) 12453

   d) 45231　　e) 6714235　　f) 31528764

2. 按照字典顺序排列下述{1,2,3,4,5,6}的排列：234561,231456,165432,156423,543216,541236,231465,314562,432561,654321,654312,435612。
3. 使用算法 1 按照字典顺序生成前 4 个正整数的 24 个排列。
4. 使用算法 2 列出集合{1,2,3,4}的所有子集。
5. 使用算法 3 列出集合{1,2,3,4,5}的所有的 3-组合
6. 证明算法 1 按字典顺序生成下一个最大的排列。
7. 证明算法 3 按字典顺序生成给定 r-组合后面的下一个最大的 r-组合。
8. 建立一个算法来生成 n 元素集合的 r-排列。
9. 列出{1,2,3,4,5}的所有 3-排列。

这一节剩下的练习建立另一个算法来生成{1,2,3,…,n}的排列。这个算法是基于整数的康托展开。每个小于 $n!$ 的非负整数有一个唯一的康托尔展开式

$$a_1 1! + a_2 2! + \cdots + a_{n-1}(n-1)!$$

其中 a_i 是一个不超过 i 的非负整数，$i=1, 2, \cdots, n-1$。整数 $a_1, a_2, \cdots, a_{n-1}$叫做这个整数的康托尔数字。

给定{1,2,…,n}的一个排列。令 a_{k-1}是排列中在 k 后面且小于 k 的整数个数，$k=2, 3, \cdots, n$。例如，在排列 43215 中，a_1 是在 2 后面且小于 2 的整数个数，所以 $a_1=1$。类似地，对这个例子 $a_2=2$，$a_3=3$，$a_4=0$。考虑从{1,2,3,…,n}的排列的集合到小于 $n!$ 的非负整数的集合的函数。这个函数把一个排列映到一个非负整数，而这个整数把以这种方式定义的 $a_1, a_2, \cdots, a_{n-1}$作为它的康托尔数字。

10. 找出对应于下述排列的整数
   a) 246531　　b) 12345　　c) 654321
*11. 证明这里描述的对应是{1,2,3,…,n}的排列的集合与小于 $n!$ 的非负整数之间的双射。
12. 按照康托尔展开式与前面练习 10 所描述的排列之间的对应找出与下面的整数相对应的{1,2,3,4,5}的排列。
   a) 3　　b) 89　　c) 111
13. 建立一个以前面练习 10 描述的对应为基础的算法来生成 n 元素集合所有的排列。
*14. 下面的方法可以用来生成一个 n 项序列的随机排列。首先，交换第 n 项与第$r(n)$项，其中$r(n)$是一个满足 $1 \leqslant r(n) \leqslant n$ 的随机选择的整数。接着，交换结果序列的第$(n-1)$项与它的第$r(n-1)$项，其中$r(n-1)$是满足$1 \leqslant r(n-1) \leqslant n-1$ 的一个随机选择的整数。继续这一过程直到 $j=n$，其中在第 j 步要交换结果序列的第$(n-j+1)$项与第$r(n-j+1)$项，其中$r(n-j+1)$是满足$1 \leqslant r(n-j+1) \leqslant n-j+1$ 的一个随机选择的整数。证明当遵循这个方法时序列项的 $n!$ 个不同排列中的每一个排列被等可能地生成[提示：使用数学归纳法。对于 $n-1$ 项的一个序列，假定这一过程生成的每一个 $n-1$ 项的排列的概率是$1/(n-1!)$。]

关键术语和结果

术语

组合数学：研究物体安排的科学

枚举：物体安排的计数

树图：由根、从根出发的分支以及从分支的某些端点出发的其他分支构成的图

排列：集合元素的一个有序的安排

r-排列：集合的 r 个元素的一个有序安排

$p(n,r)$：n 元素集合的 r-排列数

r-组合：集合的 r 个元素的无序选取

$C(n,r)$：n 元素集合的 r-组合数

$\binom{n}{r}$(二项式系数)：也是 n 元素集合的 r-组合数

帕斯卡三角形：二项式系数的一种表示，其中三角形的第 i 行包含 $C(i,j)$，$j=0$，1，2，…，i

事件的概率：该事件成功的结果次数除以可能结果的总次数

$p(E|F)$(给定条件 F 下 E 的条件概率)：$p(E\cap F)/p(F)$

独立事件：使得 $p(E\cap F)=p(E)p(F)$ 成立的事件 E 和 F

随机变量：一个函数，它对一个实验的每次结果赋一个实数值

随机变量的期望值：一个随机变量的加权平均，用结果的概率加权的随机变量的值，即 $E(X)=\sum_{s\in S}p(s)X(s)$。

随机变量的方差：随机变量的值与它的期望值之差平方的加权平均，其中的权由结果的概率给定，即 $V(X)=\sum_{s\in S}(X(s)-E(X))^2p(s)$

伯努利实验：一个具有两种可能结果的实验

结果

求和法则：一种基本的计数技术。这个法则指出，如果两个任务不能同时做，那么用这种或那种方式完成任务的总方式数是完成两种任务的方式数之和

乘积法则：一种基本的计数技术。这个法则指出，当一个过程由两个子任务构成时，完成这个过程的方式数是完成第一个任务的方式数和完成第一个任务之后再做第二个任务的方式数之积

鸽巢原理：当比 k 多的物体放到 k 个盒子时，一定存在一个盒子包含了至少 2 个物体

推广的鸽巢原理：当 N 个物体放入 k 个盒子时，一定存在一个盒子包含了至少 $\lceil N/k\rceil$ 个物体

$p(n,r)=\frac{n!}{(n-r)!}$

$C(n,r)=\frac{n!}{r!}(n-r)!$

帕斯卡恒等式：$C(n+1,k)=C(n,k-1)+C(n,k)$

二项式定理：$(x+y)^n=\sum_{k=0}^{n}C(n,k)p^kq^{n-k}$

当执行 n 次独立的伯努利实验时，k 次成功的概率等于 $C(n,k)p^kq^{n-k}$，其中 p 是成功的概率且 $q=1-p$ 是失败的概率。

当允许重复时，一个 n 元素集合的 r-排列数是 n^r。

当允许重复时，一个 n 元素集合的 r-组合数是 $C(n+r-1,r)$。

如果类型为 i 的不可区分物体有 n_i 个，$i=1, 2, 3, \cdots, k$，那么 n 个物体的排列数为

$n!(n_1!\ n_2!\ \cdots n_k!)$

生成集合 $\{1,2,\cdots,n\}$ 的排列的算法。

复习题

1. 解释怎样用求和与乘积法则找出长度不超过 10 的二进制串的个数。
2. 解释怎样找出长度不超过 10 且至少有 1 位 0 的二进制串的个数。
3. a) 怎样用乘积法则找出从 m 元素集合到 n 元素集合的函数个数？
   b) 从一个 5 元素集合到一个 10 元素集合存在多少个函数？
   c) 怎样用乘积法则找出从 m 元素集合到 n 元素集合的一对一函数的个数？
   d) 从一个 5 元素集合到一个 10 元素集合存在多少个一对一函数？
   e) 从一个 5 元素集合到一个 10 元素集合存在多少个映上的函数？
4. 如果首先赢 4 个球的队就能取胜，怎样找出两个队加赛的所有可能的结果数？
5. 怎样找出以 101 开始或以 010 结束的 10 位二进制串数？
6. a) 叙述鸽巢原理。
   b) 解释怎样用鸽巢原理证明在 11 个整数中至少两个整数的最后一位相同？
7. a) 叙述推广的鸽巢原理。
   b) 解释怎样用推广的鸽巢原理证明在 91 个整数中有 10 个整数的最后一位数字相同？
8. a) 一个 n 元素集合的 r-排列和 r-组合的区别是什么？
   b) 推导一个与 n 元素集合的 r-组合数及 r-排列数有关的等式。
   c) 有多少种方式从一班 25 个学生中选 6 个学生参加一个委员会？
   d) 有多少种方式从一班 25 个学生中选 6 个学生担任委员会中不同的常务委员？
9. a) 什么是帕斯卡三角形？
   b) 在帕斯卡三角形中的一行是怎样从它的上一行产生的？
10. 什么是恒等式的组合证明？这样的证明与代数证明有什么不同？
11. 解释怎样用组合论证证明帕斯卡恒等式。
12. a) 叙述二项式定理。
    b) 解释怎样用组合论证证明二项式定理。
    c) 求在 $(2x+5y)^{201}$ 的展开式中 $x^{100}y^{101}$ 项的系数。
13. a) 当所有的结果是等可能时定义一个事件的概率。
    b) 在买彩票时从前 50 个正整数中选择 6 个不同的中奖数，那么买一张彩票选对 6 个中奖整数的概率是多少？
14. a) 一个有限样本空间对结果的概率赋值应该满足什么条件？
    b) 如果头像出现的次数是非头像的 3 倍，那么对头像和非头像结果的概率赋值应该是多少？
15. a) 定义给定事件 F 下事件 E 的条件概率。
    b) 假设 E 是掷骰子时出现偶数点的事件，F 是掷骰子时出现 1，2 或 3 点的事件，那么给定 E 下 F 的概率是什么？
16. a) 什么时候两个事件 E 和 F 是独立的？
    b) 假设 E 是掷一个均匀的骰子时出现偶数点的事件，F 是 5 或 6 点出现的事件，那么

E 和 F 是否独立？

17. a）什么是随机变量？
    b）设 X 是随机变量，它对掷两个骰子的事件所赋的值是两个骰子上较大的点数。哪些是这个随机变量的赋值？
18. a）定义随机变量 X 的期望值。
    b）设 X 是随机变量，它对掷两个骰子的事件所赋的值是两个骰子上较大的点数。那么随机变量 X 的期望值是什么？
19. a）解释怎样把具有有限多个可能输入的算法在平均状态下的计算复杂性转变成期望值。
    b）如果要找的元素在表中的概率是 1/3，并且这个元素是表中 n 个元素中的任何一个的可能性是相等的，那么线性搜索算法在平均状态下的计算复杂性是什么？
20. a）伯努利实验的含义是什么？
    b）在 n 次独立的伯努利实验中 k 次成功的概率是多少？
    c）在 n 次独立的伯努利实验中成功次数的期望值是什么？
21. a）什么是随机变量的方差？
    b）具有成功概率为 p 的伯努利实验的方差是什么？
22. a）什么是 n 个独立随机变量的和的方差？
    b）设每次实验的成功概率为 p，当执行 n 次独立的伯努利实验时成功次数的方差是什么？
23. a）解释怎样找出与从 n 个物体允许重复的无序选取 r 个物体的方法数有关的公式。
    b）如果同种类型的物体是不加区分的，那么从 5 种不同类型的物体中选择 1 打物体有多少种方式？
    c）从这 5 种不同类型的物体中选择 12 个物体，如果第一类物体必须至少 3 个，那么有多少种方式？
    d）从这 5 种不同类型的物体中选择 12 个物体，如果第一类物体不多于 4 个，那么有多少种方式？
    e）从这 5 种不同类型的物体中选择 12 个物体，如果第一类物体必须至少 2 个，但是第二类物体不超过 3 个，那么有多少种方式？
24. a）设 n 和 r 是正整数，解释为什么方程 $x_1+x_2+\cdots+x_n=r$ 的解的个数等于 n 元素集合的允许重复的 r-组合数，这里的 x_i 是非负整数，$i=1,2,3,\cdots,n$。
    b）方程 $x_1+x_2+x_3+x_4=17$ 有多少个非负整数解？
    c）b)的方程有多少个正整数解？
25. a）n 个物体有 k 种不同的类型，其中类型 1 有 n_1 个无区别的物体，类型 2 有 n_2 个无区别的物体，…，类型 k 有 n_k 个无区别的物体，推导一个与这些物体的排列数有关的公式。
    b）有多少种方式来排序单词 *INDISCREETNESS* 的字母？
26. a）描述一个算法来生成 n 个最小正整数集合的所有排列。
27. a）把 52 张标准的扑克牌发给 6 个人，每人 5 张牌，有多少种方式？
    b）有多少种方式把 n 个有区别的物体分配给 k 个有区别的盒子且使得第 i 个盒子含有 n_i 个物体？

28. 描述一个算法来生成 n 个最小正整数集合的所有的组合？

补充练习

1. 从 10 个不同的项中选 6 项有多少种方式？
   a) 若这些项是有序选择的并且不允许重复。
   b) 若这些项是有序选择的并且允许重复。
   c) 若这些项是无序选择的并且不允许重复。
   d) 若这些项是无序选择的并且允许重复。
2. 从 6 个不同的项中选 10 项有多少种方式？
   a) 若这些项是有序选择的并且不允许重复。
   b) 若这些项是有序选择的并且允许重复。
   c) 若这些项是无序选择的并且不允许重复。
   d) 若这些项是无序选择的并且允许重复。
3. 一个考试包含 100 个真假判断题。如果答案可以空缺，一个学生回答这些考题可能有多少种不同的方式？
4. 有多少个 10 位二进制串以 000 开始或以 111 结束？
5. 字母表 $\{a,b,c\}$ 上有多少个 10 位字符串恰有 3 个 a 或恰有 4 个 b？
6. 一个校园电话系统的内部电话号码由 5 个数字组成，且第一个数字不等于 0。在这个系统中可以分配多少个不同的电话号码？
7. 一个冰激凌屋有 28 种不同口味的冰激凌，8 种不同的果汁和 12 种配料。
   a) 如果每种口味的可以不止 1 勺，并且不考虑次序，那么取 3 勺冰激凌放在一个盘中有多少种不同的方式？
   b) 如果一个小圣代包含 1 勺冰激凌、1 种果汁和 1 种配料，那么有多少种不同的小圣代？
   c) 如果一个大圣代包含 3 勺冰激凌、2 种果汁和 3 种配料。其中每种口味的冰激凌可以不止 1 个并且不考虑次序，每种果汁只能用 1 次且不考虑次序，同时每种配料也只能用 1 次并且不考虑次序。那么有多少种不同的大圣代？
8. 有多少个小于 1000 的正整数
   a) 恰有 3 个十进制数字？
   b) 有奇数个十进制数字？
   c) 至少有 1 个十进制数字等于 9？
   d) 没有奇数个十进制数字？
   e) 有两个连续的十进制数字等于 5？
   f) 是回文（即正读和倒读是一样的）？
9. 当用十进制记法写出从 1 到 1000 的数时有多少个下面的数字被用到？
   a) 0　　b) 1　　c) 2　　d) 9
10. 有黄道十二宫，需要有多少人才能保证其中至少 6 个人在同一宫？
11. 一个幸运卡甜点公司制作 213 种不同的幸运卡。一个学生使用这个公司的幸运卡在餐馆吃饭。如果这个学生没有获得 4 张同样的幸运卡，那么他在这个餐馆吃饭最多可以吃多少次？

12. 为保证至少 2 个人生在一周的同一天和同一个月（可以不在同一年），那么需要多少人？
13. 证明在 10 个不超过 50 的正整数集合中至少有 2 个不同的 5 元子集有同样的和。
14. 一包棒球卡有 20 张。如果总共有 550 种不同的卡，那么需要买多少包卡才能保证其中的 2 张卡是一样的。
15. a) 从一副牌中需要选多少张牌才能保证至少选中 2 张 A？
    b) 从一副牌中需要选多少张牌才能保证至少选中 2 张 A 和 2 种点数？
    c) 从一副牌中需要选多少张牌才能保证至少有 2 张同样点数的牌？
    d) 从一副牌中需要选多少张牌才能保证至少有 2 张不同点数的牌？

*16. 证明在任何 $n+1$ 个不超过 $2n$ 的正整数中必存在 2 个数互素。

*17. 证明在 m 个整数的序列中存在若干个连续的整数其和可被 m 整除。

18. 证明如果放 5 个点在边长为 2 的正方形中，那么其中至少有 2 个点的距离不超过$\sqrt{2}$。
19. 证明一个有理数的十进制展开式一定从某一点出现循环。
20. 一个正 n 边形有多少条对角线？这里的 n 是大于等于 3 的正整数。
21. 有多少种方式从 20 种多纳圈中选 12 个多纳圈？
    a) 如果没有 2 个多纳圈是同种的。
    b) 如果所有的多纳圈都是同种的。
    c) 如果不加限制。
    d) 如果至少有 2 种。
    e) 如果必须至少有 6 个越橘馅的多纳圈。
    f) 如果至多有 6 个越橘馅的多纳圈。
22. 从 1 到 40 之间（含 1 和 40 在内）选出 6 个连续的数作为彩票赢奖数的概率是多少？
23. 一手 13 张牌不包含对的概率是多少？
24. 求 n，如果
    a) $P(n,2)=110$　　b) $P(n,n)=5040$　　c) $P(n,4)=12P(n,2)$
25. 求 n，如果
    a) $C(n,2)=45$　　b) $C(n,3)=p(n,2)$　　c) $C(n,5)=C(n,2)$
26. 证明如果 n 和 r 是非负整数且 $n\geqslant r$，则
    $$P(n+1,r)=P(n,r)(n+1)/(n+1-r)$$
27. 给出关于$C(n,r)=C(n,n-r)$的组合证明。
28. 通过构造集合具有偶数个元素的子集与具有奇数个元素的子集之间的对应给出关于 4.3 节定理 7 的组合证明。[提示：取定集合的一个元素 a，如下构造对应：如果 a 不在子集中就把它放到子集中；如果 a 在子集中就把它从子集中取出。]
29. 设 n 和 r 是非负整数，$r<n$。证明
    $$C(n,r-1)=C(n+2,r+1)-2C(n+1,r+1)+C(n,r+1)$$
30. 使用数学归纳法证明$\sum_{j=2}^{n}C(j,2)=C(n+1,3)$，其中 n 是大于 1 的整数。
31. 使用二项式定理证明$3^n=\sum_{k=0}^{n}C(n,k)2^k$。[提示：在定理中令 $x=1$ 和 $y=2$。]
32. 在这个练习中将推导一个关于 n 个最小正整数的平方和的公式。我们将用两种方式计数三元组(i,j,k)的个数，其中 i，j 和 k 是整数且满足 $0\leqslant i<k$，$0\leqslant j<k$，$1\leqslant k<n$。
    a) 证明对于给定的 k 存在 k^2 个这样的三元组，因此有$\sum_{k=1}^{n}k^2$ 个这样的三元组。

b) 证明具有 $0 \leqslant i < j < k$ 的三元组个数和 $0 \leqslant j < i < k$ 的三元组个数都等于$C(n+1,3)$。

c) 证明具有 $0 \leqslant i = j < k$ 的三元组个数等于$C(n+1,2)$。

d) 把 a), b) 和 c) 组合起来得出

$$\sum_{k=1}^{n} k^2 = 2C(n+1,3) + C(n+1,2)$$
$$= n(n+1)(2n+1)/6$$

*33. 设 $n \geqslant 4$，有多少个 n 位二进制串恰好 01 在其中出现两次？

34. 求下述各种情况下的概率。选一手 7 张扑克牌包含

a) 2 类，其中 1 类 4 张，第 2 类的 3 张。

b) 3 类，其中 1 类 3 张，另外 2 类每类各 2 张。

c) 4 类，其中 3 类每类各 2 张，第 4 类的 1 张。

d) 5 类，其中 2 类每类 2 张，第 3，4，5 类的每类 3 张。

e) 7 类不同的牌。

f) 1 个 7 张牌的同花。

g) 1 个 7 张牌的顺子。

h) 1 个 7 张牌的同花顺子。

35. 求下述各种情况下的概率。选一手 13 张桥牌包含

a) 全部 13 张红心。

b) 同种花色的 13 张牌。

c) 7 张黑桃和 6 张梅花。

d) 一种花色的 7 张牌和另一种花色的 6 张牌。

e) 4 张方块，6 张红心，2 张黑桃和 1 张梅花。

f) 一种花色的 4 张牌，第二种花色的 6 张牌，第三种花色的 2 张牌，第四种花色的 1 张牌。

36. 设 p 和 q 是素数且 $n = pq$。随机选择小于 n 的正整数不被 p 或 q 整除的概率是多少？

*37. 设 m 和 n 是正整数。随机选择小于 mn 的正整数不被 m 或 n 整除的概率是多少？

38. 设 E_1，E_2，$\cdots E_n$ 是 n 个事件满足 $p(E_i) > 0, i = 1,2,\cdots,n$。证明

$$p(E_1 \cap E_2 \cap \cdots \cap E_n) = p(E_1)p(E_2 | E_1)p(E_3 | E_1 \cap E_2)\cdots p(E_n | E_1 \cap E_2 \cap \cdots \cap E_{n-1})$$

39. 我们说事件 E_1，E_2，$\cdots$，E_n 是相互独立的，如果

$$p(E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_m}) = p(E_{i_1})p(E_{i_2})\cdots p(E_{i_m})$$

其中 i_j 是整数，$j = 1$，2，$\cdots$，m，满足 $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$ 且 $m \geqslant 2$。

a) 写出 3 个事件 E_1，E_2，E_3 相互独立所要求的条件。

b) 令 E_1，E_2 和 E_3 分别是掷硬币第 1 次出现头像、第 2 次不出现头像和第 3 次不出现头像的事件。当一个均匀的硬币被掷 3 次时，E_1，E_2 和 E_3 相互独立吗？

c) 令 E_1，E_2 和 E_3 分别是掷硬币第 1 次出现头像、第 3 次出现头像和出现偶数个头像的事件。当一个均匀的硬币被掷 3 次时，E_1，E_2 和 E_3 相互独立吗？

d) 为证明 n 个事件是相互独立的必须检查多少个条件？

*40. 设 E 和 F 是满足$p(F) \neq 0$ 的事件。证明 E 的概率是给定 F 下 E 的概率与给定 F 的补 $\overline{F}$ 下 E 的概率的加权平均，其中的权分别是 F 和 $\overline{F}$ 的概率。即

$$p(E)=p(E|F)p(F)+p(E|\overline{F})p(\overline{F})$$

[提示：使用事实$E=(E\cap F)\cup(E\cap\overline{F})$。]

*41. 设 E 是样本空间 S 的一个事件且 F_1，F_2，…，F_n 是使得$\bigcup_{i=1}^{n}F_i=S$ 的相互排斥的事件。假定$p(E)\neq 0$ 且 $p(F_i)\neq 0, i=1,2,\cdots,n$。证明

$$p(F_j|E)=\frac{p(E|F_j)p(F_j)}{\sum_{i=1}^{n}p(E|F_i)p(F_i)}$$

[提示：使用事实 $E=\bigcup_{i=1}^{n}(E\cap F_i)$。] 这一结果叫做贝叶斯公式，因为它是被英国哲学家托马斯·贝叶斯发现的。

*42. 海王星附近的一个空间探测器使用二进制串与地球通信。假设它发送1个1用时间1/3，发送1个0用时间2/3。当发送0时，被正确接收的概率是0.9，而不被正确接收（接收成1）的概率是0.1。发送1时被正确接收的概率是0.8，而不被正确接收（接收成0）的概率是0.2。

a）用练习40找到1个0被接收的概率。

b）用练习41给出的贝叶斯公式找到给定1个0被接收的条件下0被传送的概率。

43. 设 X 是样本空间 S 的随机变量。证明$V(aX+b)=a^2V(X)$,其中 a 和 b 是实数。

44. 证明如果 m 是正整数，那么当执行每次成功概率为 p 的独立的伯努利实验时，在第 $m+n$ 次实验出现第 m 次成功的概率是$\binom{n+m-1}{n}q^np^m$。

45. 一位教授为一次离散数学考试出了20道多选题，每道题可能的答案为 a，b，c 或 d。如果具有答案 a，b，c 和 d 的试题数分别为8，3，4和5，且试题可以用任意的顺序安排，那么可能有多少种不同的答案？

46. 8个人围圆桌就座有多少种不同的安排？这里认为如果一种安排通过旋转能从另一种安排得到，那么就认为这两种安排是一样的。

47. 把24个学生分给5个指导教师有多少种方式？

48. 一蒲式耳包含20个无区别的 Delicious 苹果，20个无区别的 Macintosh 苹果，和20个无区别的 Granny Smith 苹果。从其中选12个苹果，如果每类至少选3个，有多少种方式？

49. 方程 $x_1+x_2+x_3=17$ 有多少个非负整数解？

a）若 $x_1>1$，$x_2>2$，$x_3>3$

b）若 $x_1>6$，$x_3>5$

c）若 $x_1<4$，$x_2<3$，$x_3>5$

50. 使用单词 *PEPPERCORN* 的所有字母构成字符串。

a）可以构成多少个不同的字符串？

b）其中有多少字符串以 P 开始和结束？

c）在多少个字符串中有3个连续的 P？

51. 10元素集合有多少个子集

a）少于5个元素？

b）多于7个元素？

c）有奇数个元素？

52. 一个交通逃逸事故的证人告诉警察，肇事汽车的车牌包含3个字母后面跟着3个数字，

以字母 AS 开始且包含数字 1 和 2。有多少不同的车牌符合这个描述？

53. 有多少种方式把 n 个相同的物体放入 m 个不同的容器而使得没有一个容器是空的？

54. 6 个男孩和 8 个女孩坐在一排椅子上，如果没有两个男孩相邻，有多少种方式？

55. 设计一个算法生成一个有穷集的所有允许重复的 r-排列。

56. 设计一个算法生成一个有穷集的所有允许重复的 r-组合。

计算机题目

按下述给定的输入和输出写程序。

1. 给定正整数 n 和不超过 n 的非负整数，找出 n 元素集合的 r-排列数和 r-组合数。
2. 给定正整数 n 和 r，找出 n 元素集合的允许重复的 r-排列数和允许重复的 r-组合数。
3. 给定正整数 n，找出从集合 $\{1,2,\cdots,n\}$ 选中的 6 个数就是由机械选出的中彩数的概率。
4. 给定正整数序列，找出这个序列的最长的递增和递减子序列。
5. 模拟 Monty 大厦三门问题的重复实验来计算用每种策略的赢的概率。

*6. 给定方程 $x_1+x_2+\cdots+x_n=C$，其中 C 是一个常数，x_1，x_2，$\cdots$，x_n 是非负整数，列出所有的解。

7. 给定正整数 n，按字典顺序列出集合 $\{1,2,3,\cdots,n\}$ 的所有的排列。
8. 给定正整数 n 和不超过 n 的非负整数 r，按字典顺序列出集合 $\{1,2,3,\cdots,n\}$ 的所有的 r-组合。
9. 给定正整数 n 和不超过 n 的非负整数 r，按字典顺序列出集合 $\{1,2,3,\cdots,n\}$ 的所有的 r-排列。
10. 给定正整数 n，列出集合 $\{1,2,3,\cdots,n\}$ 的所有的组合。
11. 给定正整数 n 和 r，列出集合 $\{1,2,3,\cdots,n\}$ 的允许重复的所有 r-排列。
12. 给定正整数 n 和 r，列出集合 $\{1,2,3,\cdots,n\}$ 的允许重复的所有 r-组合。
13. 给定正整数 n，生成集合 $\{1,2,3,\cdots,n\}$ 的随机排列。（见 4.7 节练习 14。）

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 当两个队加时赛时赢的队是 9 分中首先得 5 分、11 分中首先得 6 分、13 分中首先得 7 分和 15 分中首先得 8 分的队。找出加时赛的可能的结果数。
2. 哪些二项式系数是奇数？你能根据数的特征给出一个猜想吗？
3. 目前还不知道二项式系数 $C(2n,n)$ 是否一定被一个素数的平方整除，也不知道当 n 增长时在 $C(2n,n)$ 的素数分解中的最大的指数是否无界增长。通过对于尽可能多的正整数 n 找 $C(2n,n)$ 的分解式中素数的最小和最大的幂来探索这个问题。
4. 找出一手 5 张扑克牌的各种类型的概率并且根据它们的概率排列这些类型。
5. 找出在新泽西六合彩票中买 1 美元奖票有大于 1 美元的期望值的条件。为了赢奖，不管数的次序，你必须从 1 到 48 的正整数中（含 1 和 48 在内）选择被抽出的 6 个数。奖金在中奖的人中是平均分配的。必须考虑进入抽奖的奖金总额和买奖票的人数。
6. 由测试大量随机选择的整数对来估计随机选择的 2 个整数是互素的这一事件的概率。查

找给出这个概率的定理并将你的结果与正确的概率作比较。

7. 确定需要多少人才能保证其中至少 2 个人的生日在每年的同一天的概率至少是 70%，80%，90%，95%，98% 和 99%。
8. 生成 8 元素集合的所有的排列。
9. 生成 9 元素集合的所有的 6-排列。
10. 生成 8 元素集合的所有的组合。
11. 生成 7 元素集合允许重复的所有 5-组合。
12. 生成前 100 个正整数集合的 100 个随机选择的排列表（见 4.7 节练习 14）。

写作题目

用课本以外的资料，按下列要求写成短文。

1. 描述狄利克莱和其他的数学家对鸽巢原理的早期应用。
2. 讨论扩充目前电话编码计划的方式以适合对更多电话号码飞速增长的需求。（看看你是否能够找到某些来自电信产业的建议。）对你要讨论的每个新的编码计划说明怎样找到它所支持的不同电话号码的个数。
3. 本书描述了许多组合恒等式。找一找关于这种恒等式的资料，并且描述除了本书引入之外的其他重要的组合恒等式。给出其中某些恒等式的有代表性的证明，包括组合证明。
4. 描述概率论的起源和它的早期应用。
5. 描述玩轮盘赌时你可能下的不同的赌注。找出这些赌注在美国的玩法，即轮盘包含数 0 和 00 在内的概率。对你来说什么是最好的赌注？什么是最坏的赌注？
6. 讨论当你玩 21 点的纸牌游戏和 casino 纸牌游戏时赢的概率。对于在赌场下注的人是否存在一种赢的策略？
7. 描述在统计力学中的质点分布所使用的不同的模型，包括麦克斯韦—波尔兹曼、玻色—爱因斯坦和费米—狄拉克统计量，在每种情况下描述模型中使用的计数技术。
8. 定义第一类 Stirling 数并且描述它们的某些性质以及所满足的恒等式。
9. 定义第二类 Stirling 数并且描述它们的某些性质以及所满足的恒等式。
10. 定义 Ramsey 数，叙述和证明显示它们存在的 Ramsey 定理，并且描述目前已知的有关 Ramsey 数的结果。
11. 描述生成 n 元素集合所有排列的其他算法，这些算法不是在 4.7 节给出的算法。把这些算法的计算复杂性与书上和 4.7 节练习所描述算法的计算复杂性进行比较。
12. 至少描述一种方法生成一个正整数 n 的所有的剖分。（见 3.3 节练习 35。）

第5章 高级计数技术

许多计数问题用第4章讨论的方法是不容易求解的。一个这样的问题就是：有多少个 n 位二进制串不包含两个连续的0？为求解这个问题，令 a_n 是这种 n 位二进制串数。可以证明 $a_{n+1}=a_n+a_{n-1}$。这个等式叫做递推关系，它和初始条件 $a_1=2$ 和 $a_2=3$ 确定了序列 $\{a_n\}$。此外，从这个与序列的项有关的等式可以找到 a_n 的显式公式。正如我们将要看到的，可以用一种类似的技术来求解许多不同的计数问题。

我们也将看到，可以用形式幂级数也叫做生成函数来求解许多计数问题，其中 x 的幂的系数代表我们感兴趣的序列的项。除了求解计数问题，也能使用生成函数求解递推关系以及证明组合恒等式。

许多其他类型的计数问题不能使用第4章所讨论的技术求解。例如：有多少种方式把7项工作分给3个雇员而使得每个雇员至少得到一项工作？有多少个素数小于1 000？可以用计数集合并集中的元素个数来求解这两个问题。我们将建立一种技术，叫做容斥原理来计数在集合并集中的元素个数，并且将说明怎样用这种技术求解计数问题。

可以用本章学到的技术与第4章的基本技术一起求解许多计数问题。

5.1 递推关系

5.1.1 引言

一群细菌的数目每小时增加一倍。如果开始有5个细菌，在 n 小时末将有多少个细菌？为求解这个问题，令 a_n 是 n 小时末的细菌数。因为细菌数每小时增加一倍，只要 n 是正整数，关系 $a_n=2a_{n-1}$ 就成立。对所有的非负整数 n，这个关系和初始条件 $a_0=5$ 一起唯一地确定了 a_n。利用这一信息可找出关于 a_n 的公式。

某些计数问题不能用第4章给出的技术求解，但可以通过找到序列的项之间的关系，如在涉及细菌的问题中的关系，即递推关系来求解。我们将研究各种能用递推关系构造模型的计数问题。我们也将在本节和下节建立一些方法，针对满足某类递推关系的序列，求出序列的项的显式公式。

5.1.2 递推关系

在第3章我们讨论了怎样递归定义一个序列。一个序列的递归定义指定了一个或多个初始的项以及一个由前项确定后项的规则。可以用递归定义来求解计数问题。对于递归定义，这个从某些前项求后项的规则就叫做递推关系。

定义1 关于序列 $\{a_n\}$ 的递推关系是一个等式，它把 a_n 用序列中在 a_n 前面的一项或多项即 $a_0, a_1, \cdots, a_{n-1}$ 来表示，这里 $n \geqslant n_0$，n_0 是一个非负整数。如果一个序列的项满足递推关系，这个序列就叫做递推关系的解。

例1 令 $\{a_n\}$ 是一个序列，它满足递推关系 $a_n = a_{n-1} - a_{n-2}$，$n = 2, 3, 4, \cdots$，且 $a_0 = 3$，$a_1 = 5$，那么 a_2 和 a_3 是什么？

解 从递推关系可以看出，$a_2 = a_1 - a_0 = 5 - 3 = 2$ 且 $a_3 = a_2 - a_1 = 2 - 5 = -3$。 ■

例2 确定序列 $\{a_n\}$ 是否为递推关系 $a_n = 2a_{n-1} - a_{n-2}$，$n = 2, 3, 4, \cdots$，的解。这里的 $a_n = 3n$，n 是非负整数。对 $a_n = 2^n$ 和 $a_n = 5$ 也回答同一个问题。

解 假设对每一个非负整数 n，$a_n = 3n$，那么对于 $n \geq 2$，可以看出 $2a_{n-1} - a_{n-2} = 2[3(n-1)] - 3(n-2) = 3n = a_n$。于是，$\{a_n\}$ 是该递推关系的解，其中 $a_n = 3n$。

假设对每个非负整数 n，$a_n = 2^n$。注意到 $a_0 = 1$，$a_1 = 2$，$a_2 = 4$。因为 $2a_1 - a_0 = 2 \cdot 2 - 1 = 3 \neq a_2$，不难看出序列 $\{a_n\}$，$a_n = 2^n$ 不是该递推关系的解。

假设对每一个非负整数 n，$a_n = 5$，那么对于 $n \geq 2$ 有 $a_n = 2a_{n-1} - a_{n-2} = 2 \cdot 5 - 5 = 5 = a_n$。因此，$\{a_n\}$，$a_n = 5$ 是该递推关系的解。 ■

序列的初始条件说明了在递推关系起作用的首项之前的那些项。例如，例1中的 $a_0 = 3$ 和 $a_1 = 5$ 是初始条件。递推关系和初始条件唯一地确定了一个序列。这是由于一个递推关系和初始条件一起提供了这个序列的递归定义。只要使用足够多次，序列的任何一项都可以从初始条件开始通过递推关系求出。但是对于某些特定类型的序列，可以有更好的方法通过它的递推关系和初始条件来计算它的项。我们将在本节和下节讨论这些方法。

5.1.3 用递推关系构造模型

我们可以使用递推关系构造各种各样问题的模型，例如找复合利息，计数岛上的兔子，确定汉诺塔难题的移动次数，以及计数具有确定性质的二进制串。

例3 复合利息。假设一个人在银行的储蓄账上存了10 000美元，复合年息是11%。那么在30年后账上将有多少钱？

解 为求解这个问题，令 P_n 表示 n 年后的账上的钱数。因为 n 年后账上的钱等于在 $n-1$ 年后账上的钱加上第 n 年的利息，易见序列 $\{P_n\}$ 满足递推关系

$$P_n = P_{n-1} + 0.11P_{n-1} = (1.11)P_{n-1}$$

初始条件是 $P_0 = 10\ 000$。

我们可以使用迭代法找到关于 P_n 的公式。注意

$$P_1 = (1.11)P_0$$
$$P_2 = (1.11)P_1 = (1.11)^2 P_0$$
$$P_3 = (1.11)P_2 = (1.11)^3 P_0$$
$$\vdots$$
$$P_n = (1.11)P_{n-1} = (1.11)^n P_0$$

当代入初始条件 $P_0 = 10\ 000$，就得到公式 $P_n = (1.11)^n 10\ 000$。我们可以使用数学归纳法验证它的正确性。公式对 $n = 0$ 是正确的，这是初始条件的直接结果。假定 $P_n = (1.11)^n \cdot 10\ 000$，那么由递推关系和归纳假设，

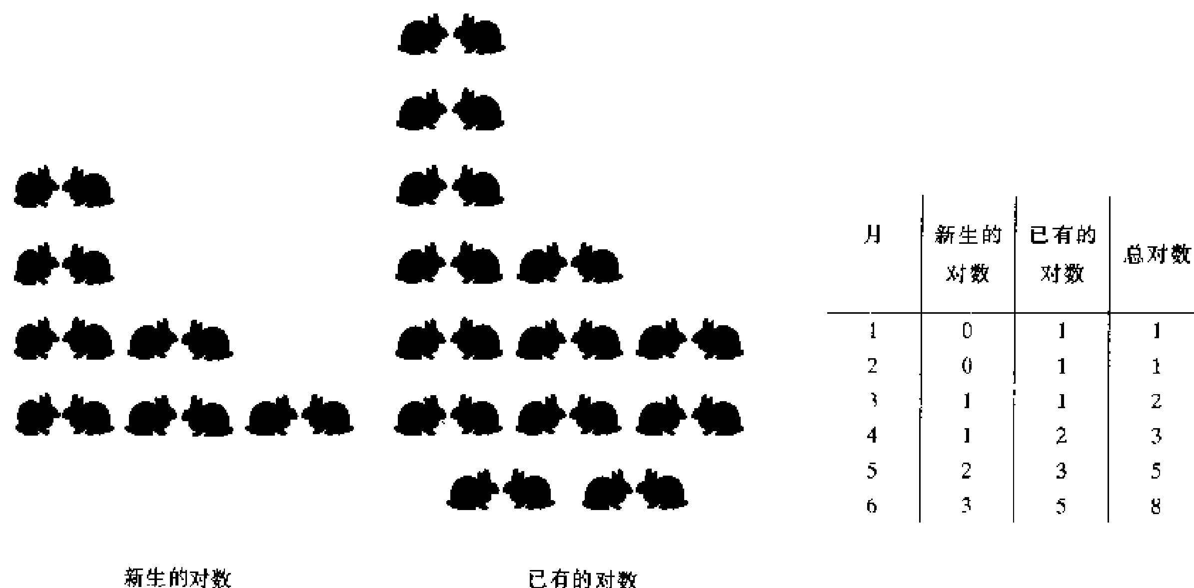
$$P_{n+1}=(1.11)P_n=(1.11)(1.11)^n 10\ 000=(1.11)^{n+1}10\ 000$$

这证明了对 P_n 的显式公式是正确的。

将 $n=30$ 代入公式 $P_n=(1.11)^n 10\ 000$ 就证明了在 30 年后账上包含 $P_{30}=(1.11)^{30}10\ 000=228\ 922.97$ 美元。■

下一个例子说明了怎样用递推关系建立关于岛上兔子数的模型。

例 4 兔子和斐波那契数。考虑下面的问题，它是由里奥那多·底·比萨，也就是斐波那契，于 13 世纪在《*Liber abaci*》一书中提出来的。一对刚出生的兔子（一公一母）被放到岛上。每对兔子出生后两个月才开始繁殖后代。如图 5-1 所示，在出生两个月以后，每对兔子在每个月都将繁殖一对新的兔子。假定兔子不会死去，找出 n 个月后关于岛上兔子对数的递推关系。



月	新生的对数	已有的对数	总对数
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8

图 5-1 岛上的兔子

解 用 f_n 表示 n 个月后的兔子对数。我们将证明 f_n，$n=1,2,3,\cdots$是斐波那契序列的项。

可以用递推关系建立兔子数的模型。在第 1 个月末，岛上的兔子对数是 $f_1=1$。由于这对兔子在第 2 个月没有繁殖，因此 $f_2=1$。为找到 n 个月后的兔子对数，要把前一个月岛上的对数 f_{n-1}加上新生的对数，而这个数等于 f_{n-2}，因为每对两个月大的兔子都生出一对新兔子。

因此，序列 $\{f_n\}$ 满足递推关系

$$f_n=f_{n-1}+f_{n-2},\quad n\geqslant 3$$

和初始条件 $f_1=1$ 和 $f_2=1$。由于这个递推关系和初始条件唯一地确定了这个序列，因此 n 个月后岛上的兔子对数由第 n 个斐波那契数给出。■

下一个例子涉及一个著名的难题。

例 5　汉诺塔。19 世纪后期一个著名的游戏叫做汉诺塔，它是由安装在一个板上的 3 根柱子和若干大小不同的盘子构成。开始时，这些盘子按照大小的次序放在第一根柱子上，使得大盘子在底下（如图 5-2 所示）。游戏的规则是：每一次把 1 个盘子从一根柱子移动到另一根柱子，但是不允许这个盘子放在比它小的盘子上面。游戏的目标是把所有的盘子按照大小的次序都放到第二根柱子上，并且将最大的盘子放在底部。

令 H_n 表示解 n 个盘子的汉诺塔问题所需要的移动次数。建立一个关于序列 $\{H_n\}$ 的递推关系。

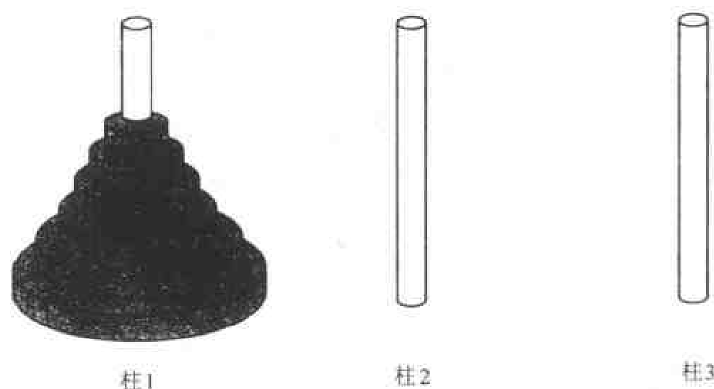


图 5-2　汉诺塔的初始位置

解　开始 n 个盘子在柱 1。按照游戏规则我们可以用 H_{n-1}次移动将上边的$n-1$个盘子移到柱 3（图 5-3 说明了此刻的柱子和盘子）。在这些移动中保留最大的盘子不动。然后，我们用一次移动将最大的盘子移到第二根柱子上。我们可以再使用 H_{n-1}次移动将柱 3 上的 $n-1$ 个盘子移到柱 2，把它们放到最大的盘子上面，这个最大的盘子一直放在柱 2 的底部。容易看出，使用更少的步数是不可能求解这个难题的。这就证明了

$$H_n = 2H_{n-1} + 1$$

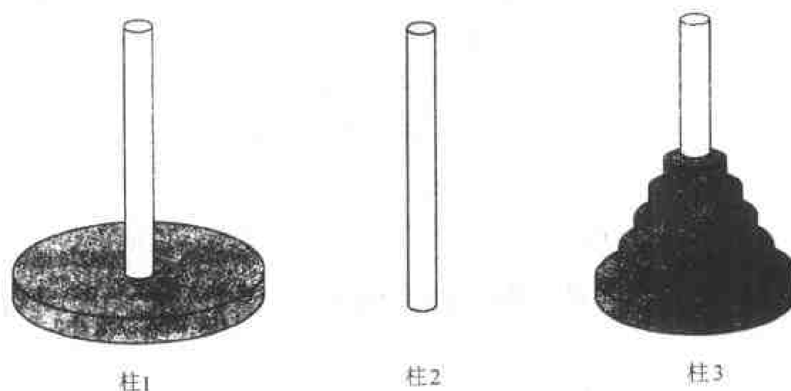


图 5-3　汉诺塔的一个中间位置

初始条件是 $H_1 = 1$，因为依照规则一个盘子可以用 1 次移动从柱 1 移到柱 2。

我们可以使用迭代方法求解这个递推关系。注意

$$\begin{aligned} H_n &= 2H_{n-1} + 1 \\ &= 2\,(2H_{n-2} + 1)\ + 1 = 2^2 H_{n-2} + 2 + 1 \end{aligned}$$

$$= 2^2(2H_{n-3}+1)+2+1 = 2^3H_{n-3}+2^2+2+1$$
$$\vdots$$
$$= 2^{n-1}H_1+2^{n-2}+2^{n-3}+\cdots+2+1$$
$$= 2^{n-1}+2^{n-2}+\cdots+2+1$$
$$= 2^n-1$$

为了用序列前面的项表示 H_n，我们重复地用到这个递推关系。在倒数第二个等式中用了初始条件 $H_1=1$。最后一个等式是基于几何级数的求和公式，这个公式可以在3.2节的例5中找到。

用迭代方法找出了具有初始条件 $H_1=1$ 的递推关系 $H_n=2H_{n-1}+1$ 的解。这个公式可以用数学归纳法证明。证明留给读者作为节后的练习。

一个古老的传说告诉我们，在汉诺有一座塔，那里的僧侣按照这个游戏的规则从一个柱子到另一个柱子移动64个金盘子。他们1秒钟移动1个盘子。据说当他们结束游戏时世界就到了末日。这个世界将在僧侣开始移动盘子多久以后终结？

根据这个显示公式，僧侣需要

$$2^{64}-1=18\ 446\ 744\ 073\ 709\ 551\ 615$$

次移动来搬这些盘子。每次移动需要1秒钟，他们将用5 000亿年来求解这个难题，因此这个世界的寿命应该比它已有的寿命更长。 ■

注意 许多人研究了源自例5所述汉诺塔难题的各种问题。某些问题用到更多的柱子，某些问题允许同样大小的盘子，某些问题对盘子的移动类型加以限制。一个最古老和最有趣的问题是雷夫难题[㊀]，它是1907年由亨利·达得尼在他的《坎特伯雷难题》(*Canterbury Puzzle*)一书中提出来的。这个难题是雷夫出的，他让一个朝圣者把一堆各种大小的乳酪从4个凳子中的一个移到另一个，移动中不允许把直径较大的乳酪放在较小的乳酪上面。如果用柱子和盘子的概念来表述雷夫难题，除了使用4根柱子之外，其他和汉诺塔的规则一样。你可能会奇怪没有人能够确定求解 n 个盘子的雷夫难题所需要的最少移动次数。但是，存在一个猜想，至今已经超过50年了。这个猜想认为所需移动的最少次数等于由富雷姆（Frame）和斯图尔特（Stewart）在1939年发明的算法所使用的移动次数。（更详细的信息可参见节末的练习48～55和[St94]。）

例6说明了递推关系怎样用于计数具有指定长度和某种性质的二进制串。

例6 对于不含2个连续0的 n 位二进制串的个数找出递推关系和初始条件。有多少个这样的5位二进制串？

解 设 a_n 表示不含2个连续0的 n 位二进制串。为得到一个关于 $\{a_n\}$ 的递推关系，由求和法则，不含2个连续0的 n 位二进制串的个数等于以0结尾的这种二进制串数加上以1结尾的这种二进制串数。我们将假定 $n\geqslant 3$，使得二进制串至少有3位。

精确地说，不含2个连续0并以1结尾的 n 位二进制串就是在不含2个连续0的 $n-1$ 位二进制串的尾部加上一个1。因此存在 a_{n-1} 个这样的二进制串。

㊀ 雷夫，Reve，更常见的是拼写为reeve，这个词在古代是指地方长官(governor)。

不含 2 个连续 0 并以 0 结尾的 n 位二进制串在它们的 $n-1$ 位必须是 1；否则它们就将以 2 个 0 结尾。因而，精确地说，不含 2 个连续 0 并以 0 结尾的 n 位二进制串就是在不含 2 个连续 0 的 $n-2$ 位二进制串的尾部加上 10。因此存在 a_{n-2} 个这样的二进制串。

如图 5-4 所示，可以断言对于 $n \geqslant 3$ 有

$$a_n = a_{n-1} + a_{n-2}$$

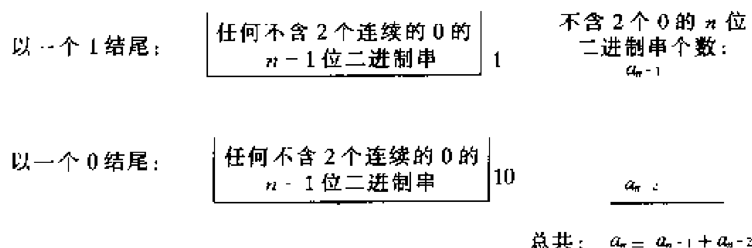


图 5－4 计数不含 2 个连续的 0 的 n 位二进制串

初始条件是 $a_1=2$，因为 1 位的二进制串是 0 和 1，没有连续的 2 个 0，而 $a_2=3$，因为 2 位的二进制串中满足条件的是 01，10 和 11。使用 3 次递推关系就可得到 a_5，

$$a_3 = a_2 + a_1 = 3+2 = 5$$

$$a_4 = a_3 + a_2 = 5+3 = 8$$

$$a_5 = a_4 + a_3 = 8+5 = 13$$

■

注意 $\{a_n\}$ 和斐波那契序列满足同样的递推关系。因为 $a_1=f_3$ 且 $a_2=f_4$，从而有 $a_n=f_{n+2}$。

下一个例子说明怎样用递推关系建立编码字数的模型，这种编码字是某些正确性检测所允许的。

例 7 编码字的枚举。一个计算机系统把一个十进制数字串作为一个编码字，如果它包含偶数个 0，就是有效的。例如，1 230 407 869 是有效的，而 120 987 045 608 不是有效的。设 a_n 是有效的 n 位编码字的个数。找出一个关于 a_n 的递推关系。

解 注意到 $a_1=9$，因为存在 10 个 1 位十进制数字串，并且只有一个，即串 0 是无效的。通过考虑怎样由 $n-1$ 位的数字串构成一个 n 位有效数字串就可以推导出关于这个序列的递推关系。从少 1 位数字的串构成 n 位有效数字串有两种方式。

第一种，在一个 $n-1$ 位的有效数字串后而加上一个非 0 的数字就可以得到一个 n 位的有效数字串。加这个数字的方式有 9 种。因此用这种方法构成 n 位有效数字串的方式有 $9a_{n-1}$ 种。

第二种，在一个无效的 $n-1$ 位数字串后面加上一个 0 就可以得到 n 位有效的数字串。(这将产生具有偶数个 0 的串，因为无效的 $n-1$ 位数字串有奇数个 0。) 这样做的方式数等于无效的 $n-1$ 位数字串的个数。因为存在 10^{n-1} 个 $n-1$ 位数字串，其中有 a_{n-1} 个是有效的，通过在无效的 $n-1$ 位数字串后而加上一个 0 就得到 $10^{n-1}-a_{n-1}$ 个 n 位的有效数字串。

因为所有的 n 位有效数字串都用这两种方式之一产生，从而存在

$$a_n = 9a_{n-1} + (10^{n-1} - a_{n-1})$$
$$= 8a_{n-1} + 10^{n-1}$$

个 n 位有效数字串。■

下面例子中的递推关系在许多不同的场合都可以见到。

例 8　求关于 C_n 的递推关系，其中 C_n 是通过对 $n+1$ 个数 $x_0, x_1, x_2, \cdots, x_n$ 的乘积中加括号来规定乘法的次序的方式数。例如，$C_3=5$，因为对 $x_0 \cdot x_1 \cdot x_2 \cdot x_3$ 有 5 种方式加括号的方式来确定乘法的次序：$((x_0 \cdot x_1) \cdot x_2) \cdot x_3$，$(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$，$(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$，$x_0 \cdot ((x_1 \cdot x_2) \cdot x_3)$ 以及 $x_0 \cdot (x_1 \cdot (x_2 \cdot x_3))$。

解　为求得关于 C_n 的递推关系，我们注意到无论怎样在 $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_n$ 中插入括号总有一个"·"运算符留在所有括号的外边，即执行最后一次乘法的运算符。[例如，在 $(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$ 中的最后一个"·"运算符，在 $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$ 中的第二个"·"运算符。] 这个最后的运算符出现在 $n+1$ 个数中的两个数之间，比如说 x_k 和 x_{k+1} 之间。当最后的运算符出现在 x_k 和 x_{k+1} 之间时，存在 C_kC_{n-k-1} 种方式插入括号来确定 $n+1$ 个数被乘的次序，因为有 C_k 种方式在乘积 $x_0 \cdot x_1 \cdot \cdots \cdot x_k$ 中插入括号来确定这 $k+1$ 个数的乘法次序，且有 C_{n-k-1} 种方式在乘积 $x_{k+1} \cdot x_{k+2} \cdot \cdots \cdot x_n$ 中插入括号来确定这 $n-k$ 个数的乘法次序。由于这个最后的运算符可能出现在 $n+1$ 个数的任两个数之间，所以

$$C_n = C_0C_{n-1} + C_1C_{n-2} + \cdots + C_{n-2}C_1 + C_{n-1}C_0$$
$$= \sum_{k=0}^{n-1} C_kC_{n-k-1}$$

注意到初始条件是 $C_0=1$ 和 $C_1=1$。这个递推关系可以用生成函数的方法求解，这种方法将在 5.4 节讨论。可以证明 $C_n = C(2n, n)/(n+1)$。(见节末的练习 41。) ■

序列 $\{C_n\}$ 是卡特朗 (Catalan) 数的序列。这个序列是除此之外的许多不同计数问题的解（细节见 [MiRo90] 或 [Rob84] 中有关卡特朗数的一章)。

练习

1. 求给定序列的前五项，每个序列由下面的递推关系和初始条件定义。
   a) $a_n = 6a_{n-1}$，$a_0=2$
   b) $a_n = a_{n-1}^2$，$a_1=2$
   c) $a_n = a_{n-1} + 3a_{n-2}$，$a_0=1$，$a_1=2$
   d) $a_n = na_{n-1} + n^2a_{n-2}$，$a_0=1$，$a_1=1$
   e) $a_n = a_{n-1} + a_{n-3}$，$a_0=1$，$a_1=2$，$a_2=0$
2. 证明序列 $\{a_n\}$ 是递推关系 $a_n = -3a_{n-1} + 4a_{n-2}$ 的解，如果
   a) $a_n = 0$
   b) $a_n = 1$
   c) $a_n = (-4)^n$

d) $a_n=2(-4)^n+3$

3. 序列 $\{a_n\}$ 是递推关系 $a_n=8a_{n-1}-16a_{n-2}$ 的解吗？如果

a) $a_n=0$

b) $a_n=1$

c) $a_n=2^n$

d) $a_n=4^n$

e) $a_n=n4^n$

f) $a_n=2\cdot 4^n+3n\cdot 4^n$

g) $a_n=(-4)^n$

h) $a_n=n^2 4^n$

4. 对下面每一个序列求出满足这个序列的递推关系。(答案不是唯一的，因为任何序列满足的递推关系有无数多个。)

a) $a_n=3$

b) $a_n=2n$

c) $a_n=2n+3$

d) $a_n=5^n$

e) $a_n=n^2$

f) $a_n=n^2+n$

g) $a_n=n+(-1)^n$

h) $a_n=n!$

5. 用例 5 中的迭代方法求下面每个递推关系和初始条件的解。

a) $a_n=3a_{n-1}$, $a_0=2$

b) $a_n=a_{n-1}+2$, $a_0=3$

c) $a_n=a_{n-1}+n$, $a_0=1$

d) $a_n=a_{n-1}+2n+3$, $a_0=4$

e) $a_n=2a_{n-1}-1$, $a_0=1$

f) $a_n=3a_{n-1}+1$, $a_0=1$

g) $a_n=na_{n-1}$, $a_0=5$

h) $a_n=2na_{n-1}$, $a_0=1$

6. 一个人在账上存入 1 000 美元，每年的复利是 9%。

a) 对于 n 年后账上的钱数建立一个递推关系。

b) 对于 n 年后账上的钱数求出一个显示公式。

c) 在 100 年以后账上将有多少钱？

7. 假设一群细菌的数目每小时增长为 3 倍。

a) 建立关于 n 小时后细菌数的递推关系。

b) 如果初始的群体有 100 个细菌，那么 10 小时后将有多少个细菌？

8. 假设世界人口在 1999 年是 60 亿，每年的增长率为 1.3%。

web a) 对于 1999 年后 n 年的世界人口建立一个递推关系。

b) 求出 1999 年后 n 年的世界人口的显示公式。

c) 在2020年世界的人口将是多少？

9. 一个工厂逐月增长地定做体育赛车。在第1个月只做了1辆，在第2个月做了2辆，照此下去，到第n个月做了n辆。

a) 对这个工厂前n个月生产的赛车数构造一个递推关系。

b) 在第一年生产了多少赛车？

c) 求出这个工厂在前n个月生产赛车数的显示公式。

10. 一个雇员在1987年进入一个公司，初始的年薪是50 000美元。每年这个雇员的收入增加1 000美元外加前一年工资的5%。

a) 对这个雇员在1987年后n年的工资建立一个递推关系。

b) 这个雇员在1995年的工资是多少？

c) 求出这个雇员在1987年后n年工资的显示公式。

11. 用数学归纳法验证在例5导出的求解汉诺塔难题所需移动次数的公式。

12. a) 找到一个关于n元素集合的排列数的递推关系。

b) 通过迭代用这个递推关系求n元素集合的排列数。

13. 一台出售邮票簿的售货机只接受1美元硬币、1美元纸币以及5美元纸币。

a) 找出与放n美元到这台售货机的方式数有关的递推关系，这里要考虑硬币和纸币放入的次序。

b) 初始条件是什么？

c) 一本邮票簿需10美元，有多少种方式付款？

14. 一个国家使用的硬币价值为1比索、2比索、5比索、10比索，纸币的价值为5比索、10比索、20比索、50比索和100比索。如果考虑付硬币和纸币的次序，求一个与付n比索账单的方式数有关的递推关系。

15. 如果考虑付硬币和纸币的次序，那么使用练习14描述的货币系统付17比索的账单有多少种方式？

*16. a)设n是正整数，求一个与下述正整数序列的个数有关的递推关系。这种序列要以1作为首项，以n作为末项并且是严格递增的。即序列$a_1, a_2, \cdots, a_k$，其中$a_1=1$，$a_k=n$，且对$j=1, 2, \cdots, k-1$，$a_j<a_{j+1}$。

b) 初始条件是什么？

c) 当n是大于等于2的正整数时，有多少个a)中所描述的序列？

17. a) 求与包含2个连续0的n位二进制串的个数有关的递推关系。

b) 初始条件是什么？

c) 包含2个连续0的7位二进制串有多少个？

18. a) 求与包含3个连续0的n位二进制串的个数有关的递推关系。

b) 初始条件是什么？

c) 包含3个连续0的7位二进制串有多少个？

19. a) 求与不包含3个连续0的n位二进制串的个数有关的递推关系。

b) 初始条件是什么？

c) 不包含3个连续0的7位二进制串有多少个？

*20. a）求与包含 01 的 n 位二进制串的个数有关的递推关系。

b）初始条件是什么？

c）包含 01 的 7 位二进制串有多少个？

21. a）一个人爬阶梯，如果每次可以上 1 或 2 阶，求与爬 n 步阶梯的方式数有关的递推关系。

b）初始条件是什么？

c）这个人爬 8 步阶梯上飞机有多少种方式？

22. a）如果一个人爬阶梯每次可以上 1，2 或 3 阶，求与爬 n 步阶梯的方式数有关的递推关系。

b）初始条件是什么？

c）这个人爬 8 步阶梯上飞机有多少种方式？

一个只包含 0，1 和 2 的串叫做三进制串。

23. a）求与不包含 2 个连续 0 的 n 位三进制串的个数有关的递推关系。

b）初始条件是什么？

c）不包含 2 个连续 0 的 6 位三进制串有多少个？

24. a）求与包含 2 个连续 0 的 n 位三进制串的个数有关的递推关系。

b）初始条件是什么？

c）包含 2 个连续 0 的 6 位三进制串有多少个？

*25. a）求与不包含 2 个连续 0 或 2 个连续 1 的 n 位三进制串的个数有关的递推关系。

b）初始条件是什么？

c）不包含 2 个连续 0 或 2 个连续 1 的 6 位三进制串有多少个？

*26. a）求与包含 2 个连续 0 或 2 个连续 1 的 n 位三进制串的个数有关的递推关系。

b）初始条件是什么？

c）包含 2 个连续 0 或 2 个连续 1 的 6 位三进制串有多少个？

*27. a）求与不包含连续的相同符号的 n 位三进制串个数有关的递推关系。

b）初始条件是什么？

c）不包含连续的相同符号的 6 位三进制串有多少个？

**28. a）求包含 2 个连续的相同符号的 n 位三进制串个数的递推关系。

b）初始条件是什么？

c）包含 2 个连续的相同符号的 6 位三进制串有多少个？

29. 信息通过信道传送要使用两个信号。一个信号的传送需要 1 微秒，而另一个信号的传送需要 2 微秒。

a）求与在 n 微秒内发送的不同信息数有关的递推关系，其中信息由这两个信号的序列构成，并且信息中的每个信号后面都紧跟着下一个信号。

b）初始条件是什么？

c）用这两个信号在 10 微秒内可以发送多少条不同的信息？

30. 一个汽车司机只用 5 美分和 10 美分硬币付过桥费，每次向收费机投一个硬币。

a）求与这个汽车司机付费 n 美分的不同方式数有关的递推关系（考虑使用硬币的次序）。

b）这个司机付费45美分有多少种可能的方式？

31. a）找出由 R_n 满足的递推关系，其中 R_n 是一个平面被 n 条直线划分的区域个数，如果没有两条直线是平行的也没有3条直线交于一点。

b）使用迭代求出 R_n。

*32. a）找出由 R_n 满足的递推关系，其中 R_n 是一个球面被 n 个大圆（球面与通过球心的平面的交线）划分的区域个数，如果没有3个大圆交于一点。

b）使用迭代求出 R_n。

*33. a）找出由 S_n 满足的递推关系，其中 S_n 是三维空间被 n 个平面分成的区域数，如果每3个平面交于一点，但没有4个平面交于一点。

b）使用迭代求出 S_n。

34. 求出与具有偶数个0的 n 位二进制串个数有关的递推关系。

35. 包含偶数个0的7位二进制串有多少个？

36. a）找到与用1×2的多米诺牌完全覆盖2×n 的棋盘的方式数有关的递推关系。[提示：分别考虑对棋盘右上角的位置用一张多米诺牌水平放置和垂直放置的覆盖的方式。]

b）关于a）中递推关系的初始条件是什么？

c）用1×2的多米诺牌完全覆盖2×17的棋盘有多少种方式？

37. a）用地砖铺一条人行道，地砖是红色、绿色或灰色的。如果没有两块红砖相邻且同色的地砖是不加区别的，找出与用 n 块砖铺一条路的方式数有关的递推关系。

b）对于a）中的递推关系有什么初始条件？

c）用7块砖铺一条在a）中所描述的路有多少种方式？

38. 证明斐波那契数满足递推关系 $f_n=5f_{n-4}+3f_{n-5}$，$n=5, 6, 7, \cdots$，其中这个递推关系具有初始条件 $f_0=0$，$f_1=1$，$f_2=1$，$f_3=2$，$f_4=3$。用这个递推关系证明 f_{5n} 可被5整除，$n=1, 2, 3, \cdots$。

*39. 设 $S(m, n)$ 表示从 m 元素集合到 n 元素集合的映上函数个数。证明 $S(m,n)$ 满足递推关系

$$S(m,n) = n^m - \sum_{k=1}^{n-1} C(n,k)S(m,k)$$

其中 $m \geqslant n$ 且 $n>1$，初始条件是 $S(m,1)=1$。

40. a）写出为确定相乘次序而在乘积 $x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4$ 中加括号的所有的方式。

b）使用在例8所建立的递推关系计算 C_4，即为确定相乘的次序在5个数的乘积中加括号的方式数。验证你在a）列出的方式数是正确的。

c）使用在例8的解答中所提到的关于 C_n 的封闭公式，通过求出 C_4 检验你在b）得到的结果。

41. a）使用在例8所建立的递推关系确定 C_5，即为确定相乘的次序在6个数的乘积中加括号的方式数。

b）使用在例8的解答中所提到的关于 C_5 的封闭公式检验你在b）得到的结果。

*42. 在汉诺塔难题中，假设我们的目标是把所有的 n 个盘子从柱1移到柱3，但我们不能直接在柱1和柱3之间移动盘子。每次移动盘子必须通过柱2。和通常的汉诺塔问题一

样，我们不能把较大的盘子放在较小的盘子上面。

a）找出与求解这个具有附加限制条件的 n 个盘子的难题所需移动次数有关的递推关系。

b）解这个递推关系来确定求解这个 n 个盘子难题所需移动次数的公式。

c）有多少种不同的方法把 n 个盘子安排在 3 个柱子上使得没有一个较大的盘子放在较小的盘子上面？

d）显示在这个变形难题的解中得到的对 n 个盘子的各种可能的安排。

练习 43－47 是格雷厄姆·克努斯（Graham Knuth）和帕塔什尼克（Patashnik）在[GrKnPa94]所描述的约瑟夫问题的一种变形。这个问题来源于历史学家弗劳瓦斯·约瑟夫的一本账。41 个犹太叛民在一世纪犹太－罗马战争期间被罗马人追赶逃人山洞，约瑟夫是这群人的中的一个。这些叛民宁愿死也不愿被捕；他们决定围成一个圆圈并且围着这个圆圈重复数数，每数到 3 就杀掉这个位置的人而留下其他的人。但是约瑟夫和另一个叛民不愿意就这样被杀掉；他们确定了他们应该站的位置是最后两个活下来的叛民的位置。我们考虑的问题开始时有 n 个人，记为 1 到 n，站成一个圆圈。每一步，每第 2 个仍旧活着的人将被排除，直到只剩下一个人为止。我们把生还的人数记作 $J(n)$。

43. 对每个正整数 n 的值，$1 \leqslant n \leqslant 16$，确定 $J(n)$ 的值。
44. 使用你在练习 43 找到的值猜想一个关于 $J(n)$ 的公式。［提示：写 $n = 2^m + k$，其中 m 是非负整数，k 是小于 2^m 的非负整数。］
45. 对于 $n \geqslant 1$，证明 $J(n)$ 满足递推关系 $J(2n) = 2J(n) - 1$ 和 $J(2n+1) = 2J(n) + 1$，且 $J(1) = 1$。
46. 用练习 45 的递推关系根据数学归纳法证明你在练习 44 所猜想的公式。
47. 根据你关于 $J(n)$ 的公式确定 $J(100)$，$J(1\ 000)$ 和 $J(10\ 000)$。

练习 48～55 涉及雷夫难题，即具有 4 个柱和 n 个盘子的汉诺塔的变形问题。在给出这些练习之前，我们描述一个富雷姆－斯图尔特（Frame－Stewart）算法，它把盘子从柱 1 移到柱 4 并且没有较大的盘子放在较小的盘子上面。给定盘子数 n 作为输人，这个算法依赖于一个整数 k 的选择，$1 \leqslant k \leqslant n$。当只有一个盘子时，把它从柱 1 移到柱 4，然后算法停止。对于 $n > 1$，算法递归地使用下面的 3 步。首先使用所有的 4 根柱递归地把最小的 $n-k$ 个盘子从柱 1 移到柱 2。下一步使用汉诺塔问题的三根柱算法，不使用放 $n-k$ 个最小盘子的柱，把 k 个最大的盘子递归地从柱 1 移到柱 4。最后，使用所有 4 根柱递归地将 $n-k$ 个最小的盘子移到柱 4。富雷姆和斯图尔特证明，使用他们的算法，为了达到最少的移动次数，应该选择 k 使得 n 是不超过第 k 个三角形数 $t_k = k(k+1)/2$ 的最小的正整数，即 $t_{k-1} < n \leqslant t_k$。有一个未被证实的猜想，称为富雷姆猜想，就是不管盘子怎样移动，该算法对于求解这个难题所需要的移动次数最少。

48. 证明具有 3 个盘子的雷夫难题最少可以使用 5 次移动求解。
49. 证明具有 4 个盘子的雷夫难题最少可以使用 9 次移动求解。
50. 描述富雷姆－斯图尔特算法所做的移动，并选择 k 使得在下面每种情况下所需要的移动次数最少。

    a）5 个盘子　　b）6 个盘子　　c）7 个盘子　　d）8 个盘子

*51. 证明如果$R(n)$是由富雷姆－斯图尔特算法求解具有 n 个盘子的雷夫难题所使用的移动次数，这里选择 k 是满足$n \leqslant k(k+1)/2$ 的最小的整数，那么 $R(n)$ 满足递推关系 $R(n)=2R(n-k)+2^k-1$ 和 $R(0)=0, R(1)=1$。

*52. 证明如果 k 如练习 51 所选，那么$R(n)-R(n-1)=2^{k-1}$。

*53. 证明如果 k 如练习 51 所选，那么$R(n)=\sum_{i=1}^{k} i2^{i-1}-(t^k-n)2^{k-1}$。

*54. 用练习 53 给出对所有的整数 n，$1 \leqslant n \leqslant 25$，求解雷夫难题所需移动次数的上界。

*55. 证明$R(n)$是 $O(\sqrt{n}2^{\sqrt{2n}})$。

设 $\{a_n\}$ 是实数序列，这个序列的向后差分递归地定义如下：

首项差分∇a_n 是

$$\nabla a_n = a_n - a_{n-1}$$

从$\nabla^k a_n$ 得到第 $k+1$ 项差分$\nabla^{k+1} a_n$，即

$$\nabla^{k+1} a_n = \nabla^k a_n - \nabla^k a_{n-1}$$

56. 求关于序列$\{a_n\}$的∇a_n，其中

a) $a_n=4$　　b) $a_n=2n$　　c) $a_n=n^2$　　d) $a_n=2^n$

57. 对于在练习 34 中的序列求$\nabla^2 a_n$。

58. 证明 $a_{n-1}=a_n-\nabla a_n$。

59. 证明 $a_{n-2}=a_n-2\nabla a_n+\nabla^2 a_n$。

*60. 证明a_{n-k}可以用$a_n, \nabla a_n, \nabla^2 a_n, \cdots, \nabla^k a_n$ 的项表示。

61. 用$a_n, \nabla a_n, \nabla^2 a_n$ 的项表示递推关系$a_n=a_{n-1}+a_{n-2}$。

62. 证明关于序列$\{a_n\}$的任何递推关系都可以用$a_n, \nabla a_n, \nabla^2 a_n, \cdots$的项表示。涉及这个序列和它的差分的等式叫做*差分方程*。

5.2 求解递推关系

5.2.1 引言

各种各样的递推关系出现在模型里。某些递推关系可以用迭代或者其他的特别技术求解。但是，有一类重要的递推关系可以用一种系统的方法明确地求解。在这种递推关系中，序列的项由它的前项的线性组合来表示。

定义 1 一个常系数的 k 阶线性齐次递推关系是形如

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

的递推关系，其中 c_1，c_2，$\cdots$，c_k 是实数，$c_k \neq 0$。

这个定义中的递推关系是线性的，因为它的右边是序列前项的倍数之和。这个递推关系是齐次的，因为所出现的各项都是 a_j 的倍数。序列各项的系数都是常数而不是依赖于 n 的函数。阶为 k 是因为a_n 由序列前面的k 项来表示。

根据第二数学归纳原理，满足这个定义的递推关系的序列由这个递推关系和 k 个初始条件

$$a_0 = C_0,\ a_1 = C_1,\ \cdots,\ a_{k-1} = C_{k-1}$$

唯一地确定。

例 1　递推关系 $P_n = (1.11)P_{n-1}$ 是 1 阶的线性齐次递推关系。递推关系 $f_n = f_{n-1} + f_{n-2}$ 是 2 阶的线性齐次递推关系。递推关系 $a_n = a_{n-5}$ 是 5 阶的线性齐次递推关系。■

下面是一些常系数的但不是线性齐次递推关系的例子。

例 2　递推关系 $a_n = a_{n-1} + a_{n-2}^2$ 不是线性的。递推关系 $H_n = 2H_{n-1} + 1$ 不是齐次的。递推关系 $B_n = nB_{n-1}$ 不是常系数的。■

研究线性齐次递推关系有两个理由。第一，在建立问题的模型时经常出现这种递推关系。第二，它们可以用系统的方法求解。

5.2.2　求解常系数线性齐次递推关系

求解常系数线性齐次递推关系的基本方法是寻找形如 $a_n = r^n$ 的解，其中 r 是常数。注意 $a_n = r^n$ 是递推关系 $a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k}$ 的解，当且仅当

$$r^n = c_1r^{n-1} + c_2r^{n-2} + \cdots + c_kr^{n-k}$$

当等式的两边除以 r^{n-k} 并且从右边减去左边时，我们得到等价的方程

$$r^k - c_1r^{k-1} - c_2r^{k-2} - \cdots - c_{k-1}r - c_k = 0$$

因此，序列 $\{a_n\}$ 以 $a_n = r^n$ 作为解，当且仅当 r 是这后一个方程的解。这个方程叫做该递推关系的*特征方程*。方程的解叫做该递推关系的*特征根*。正如我们将要看到的，可以用这些特征根给出这种递推关系的所有解的显示公式。

我们首先看一个 2 阶常系数线性齐次递推关系的处理结果。然后，叙述相应的阶可能大于 2 的一般性结果。由于得到一般性结果所需要的证明比较复杂，本书不再赘述。

我们现在回到 2 阶线性齐次递推关系。首先，考虑存在两个不等的特征根的情况。

定理 1　设 c_1 和 c_2 是实数。假设 $r^2 - c_1r - c_2 = 0$ 有两个不等的根 r_1 和 r_2，那么序列 $\{a_n\}$ 是递推关系 $a_n = c_1a_{n-1} + c_2a_{n-2}$ 的解，当且仅当 $a_n = \alpha_1r_1^{\,n} + \alpha_2r_2^{\,n}$，$n = 0, 1, 2, \cdots$，其中 α_1 和 α_2 是常数。

证　证明这个定理必须做两件事。首先，必须证明如果 r_1 和 r_2 是特征方程的根，并且 α_1 和 α_2 是常数，那么序列 $\{a_n\}$ $(a_n = \alpha_1r_1^{\,n} + \alpha_2r_2^{\,n})$ 是递推关系的解。第二，必须证明如果序列 $\{a_n\}$ 是解，那么对某个常数 α_1 和 α_2 有 $a_n = \alpha_1r_1^{\,n} + \alpha_2r_2^{\,n}$。

现在我们将证明如果 $a_n = \alpha_1r_1^{\,n} + \alpha_2r_2^{\,n}$，那么序列 $\{a_n\}$ 是递推关系的解。因为 r_1 和 r_2 是 $r^2 - c_1r - c_2 = 0$ 的根，从而 $r_1^{\,2} = c_1r_1 + c_2$，$r_2^{\,2} = c_1r_2 + c_2$。

从这些等式可以看出

$$\begin{aligned} c_1a_{n-1} + c_2a_{n-2} &= c_1(\alpha_1r_1^{\,n-1} + \alpha_2r_2^{\,n-1}) + c_2(\alpha_1r_1^{\,n-2} + \alpha_2r_2^{\,n-2}) \\ &= \alpha_1r_1^{\,n-2}(c_1r_1 + c_2) + \alpha_2r_2^{\,n-2}(c_1r_2 + c_2) \\ &= \alpha_1r_1^{\,n-2}r_1^{\,2} + \alpha_2r_2^{\,n-2}r_2^{\,2} \end{aligned}$$

$$= \alpha_1 r_1^n + \alpha_2 r_2^n$$
$$= a_n$$

这证明了序列$\{a_n\}$($a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$)是递推关系的解。

为证明递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的每一个解$\{a_n\}$都有形式 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$，$n = 0, 1, 2, \cdots$，$\alpha_1$ 和 α_2 为某个常数，假设$\{a_n\}$是递推关系的解，初始条件是 $a_0 = C_0$，$a_1 = C_1$。下面证明存在常数 α_1 和 α_2 使得具有 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ 的序列 $\{a_n\}$ 满足同样的初始条件。这要求

$$a_0 = C_0 = \alpha_1 + \alpha_2$$
$$a_1 = C_1 = \alpha_1 r_1 + \alpha_2 r_2$$

我们可以求解这两个关于 α_1 和 α_2 的方程。从第一个方程得到 $\alpha_2 = C_0 - \alpha_1$。把它代入第二个方程得

$$C_1 = \alpha_1 r_1 + (C_0 - \alpha_1) r_2$$

因此，

$$C_1 = \alpha_1 (r_1 - r_2) + C_0 r_2$$

这说明了

$$\alpha_1 = \frac{(C_1 - C_0 r_2)}{r_1 - r_2}$$

和

$$\alpha_2 = C_0 - \alpha_1 = C_0 - \frac{(C_1 - C_0 r_2)}{r_1 - r_2} = \frac{C_0 r_1 - C_1}{r_1 - r_2}$$

这里关于 α_1 和 α_2 的表达式依赖于 $r_1 \neq r_2$ 的事实。(当 $r_1 = r_2$ 时，这个定理不成立。)因此，由于这两个 α_1 和 α_2 的值，具有 $\alpha_1 r_1^n + \alpha_2 r_2^n$ 的序列$\{a_n\}$满足这两个初始条件。因为这个递推关系和这些初始条件唯一地确定了这个序列，从而得出 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$。 □

常系数线性齐次递推关系的特征根可能是复数。定理 1（和本节后面的几个定理）在这种情况下仍旧适用。具有复数特征根的递推关系在这本书中不加讨论。熟悉复数的读者可以做节末的练习 38 和 39。

下面的例子说明定理 1 给出的公式是很有用的。

例 3 什么是下面递推关系的解？

$$a_n = a_{n-1} + 2a_{n-2}$$

其中 $a_0 = 2$ 和 $a_1 = 7$。

解 可用定理 1 求解这个问题。递推关系的特征方程是 $r^2 - r - 2 = 0$。它的根是 $r = 2$ 和 $r = -1$。因此，序列$\{a_n\}$是递推关系的解，当且仅当

$$a_n = \alpha_1 2^n + \alpha_2 (-1)^n$$

α_1 和 α_2 是常数。由初始条件，得

$$a_0 = 2 = \alpha_1 + \alpha_2$$
$$a_1 = 7 = \alpha_1 \cdot 2 + \alpha_2 \cdot (-1)$$

求解这两个等式得 $\alpha_1 = 3$ 和 $\alpha_2 = -1$。于是，关于这个递推关系和初始条件的解是序列 $\{a_n\}$，其中

$$a_n = 3 \cdot 2^n - (-1)^n$$ ■

例 4　找一个关于斐波那契数的显示公式。

解　斐波那契数的序列满足递推关系 $f_n = f_{n-1} + f_{n-2}$ 和初始条件 $f_0 = 0$ 和 $f_1 = 1$。特征方程 $r^2 - r - 1 = 0$ 的根是 $r_1 = (1+\sqrt{5})/2$ 和 $r_2 = (1-\sqrt{5})/2$。因此，从定理 1 得到斐波那契数由

$$f_n = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

给出，其中 α_1 和 α_2 为常数。可用初始条件 $f_0 = 0$ 和 $f_1 = 1$ 确定这些常数。我们有

$$f_0 = \alpha_1 + \alpha_2 = 0$$
$$f_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1$$

对这些关于 α_1 和 α_2 的联立方程的解是

$$\alpha_1 = 1/\sqrt{5}, \quad \alpha_2 = -1/\sqrt{5}$$

于是，斐波那契数由下面的式子给出：

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$ ■

当存在二重特征根时定理 1 不再适用。这种情况可使用下面的定理来处理。

定理 2　设 c_1 和 c_2 是实数，$c_2 \neq 0$。假设 $r^2 - c_1 r - c_2 = 0$ 只有一个根 r_0。序列 $\{a_n\}$ 是递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的解，当且仅当 $a_n = \alpha_1 r_0{}^n + \alpha_2 n r_0{}^n$，$n = 0, 1, 2, \cdots$，其中 α_1 和 α_2 是常数。

定理 2 的证明留作节末的练习。下面的例子说明了这个定理的应用。

例 5　具有初始条件 $a_0 = 1$ 和 $a_1 = 6$ 的递推关系

$$a_n = 6a_{n-1} - 9a_{n-2}$$

的解是什么？

解　$r^2 - 6r + 9 = 0$ 的唯一的根是 $r = 3$。因此，这个递推关系的解是：

$$a_n = \alpha_1 3^n + \alpha_2 n 3^n$$

其中 α_1 和 α_2 是常数。使用初始条件得

$$a_0 = 1 = \alpha_1$$
$$a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$$

求解这两个方程得 $\alpha_1=1$ 和 $\alpha_2=1$。从而，这个具有给定初始条件的递推关系的解是

$$a_n=3^n+n3^n$$

■

我们现在叙述这个关于常系数线性齐次递推关系的解的一般性结果，这里的阶可以大于2且假定特征方程有不等的根。这个结果的证明给读者留作练习。

定理 3 设 c_1，c_2，…，c_k 是实数。假设特征方程

$$r^k-c_1r^{k-1}-\cdots-c_k=0$$

有 k 个不等的根 r_1，r_2，…，r_k 那么序列 $\{a_n\}$ 是递推关系

$$a_n=c_1a_{n-1}+c_2a_{n-2}+\cdots+c_ka_{n-k}$$

的解，当且仅当

$$a_n=\alpha_1r_1{}^n+\alpha_2r_2{}^n+\cdots+\alpha_kr_k{}^n$$

$n=0$，1，2，…，其中 α_1，α_2，…，α_k 是常数。

我们用例子说明定理的使用。

例 6 求出具有初始条件 $a_0=2$，$a_1=5$ 和 $a_2=15$ 的递推关系

$$a_n=6a_{n-1}-11a_{n-2}+6a_{n-3}$$

的解。

解 这个递推关系的特征多项式是

$$r^3-6r^2+11r-6$$

因为$r^3-6r^2+11r-6=(r-1)(r-2)(r-3)$,所以特征根是 $r=1$，$r=2$ 和 $r=3$。因此，递推关系的解的形式是

$$a_n=\alpha_1\cdot1^n+\alpha_2\cdot2^n+\alpha_3\cdot3^n$$

为找到常数 α_1，α_2 以及 α_3，使用初始条件得

$$a_0=2=\alpha_1+\alpha_2+\alpha_3$$
$$a_1=5=\alpha_1+\alpha_2\cdot2+\alpha_3\cdot3$$
$$a_2=15=\alpha_1+\alpha_2\cdot4+\alpha_3\cdot9$$

当求解这三个关于 α_1，α_2，α_3 的联立方程时，得到 $\alpha_1=1$，$\alpha_2=-1$，$\alpha_3=2$。于是，这个递推关系和给定初始条件的唯一解是满足

$$a_n=1-2^n+2\cdot3^n$$

的序列 $\{a_n\}$。 ■

我们现在叙述关于常系数线性齐次递推关系的最一般化的结果，这里允许特征方程有重根。要点是对于特征方程的每个根 r，通解是形如$P(n)r^n$ 的项之和，其中$P(n)$是 $m-1$次多项式，而 m 是这个根的重数。我们把证明作为一个挑战性的练习留给读者。

定理 4 设 c_1，c_2，…，c_k 是实数，假设特征方程

$$r^k - c_1 r^{k-1} - \cdots - c_k = 0$$

有 t 个不等的根 r_1，$r_2 \cdots$，r_t，其重数分别为 m_1，m_2，…，m_t，满足 $m_i \geqslant 1$，$i = 1, 2, \cdots, t$，且 $m_1 + m_2 + \cdots + m_t = k$。那么序列 $\{a_n\}$ 是递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

的解，当且仅当

$$\begin{aligned} a_n = &(\alpha_{1,0} + \alpha_{1,1} n + \cdots + \alpha_{1,m_1-1} n^{m_1-1}) r_1^n \\ &+ (\alpha_{2,0} + \alpha_{2,1} n + \cdots + \alpha_{2,m_2-1} n^{m_2-1}) r_2^n \\ &+ \cdots + (\alpha_{t,0} + \alpha_{t,1} n + \cdots + \alpha_{t,m_t-1} n^{m_t-1}) r_t^n \end{aligned}$$

$n = 0, 1, 2, \cdots$，其中 $\alpha_{i,j}$ 是常数，$1 \leqslant i \leqslant t$ 且 $0 \leqslant j \leqslant m_i - 1$。

下面的例子说明在特征方程有重根时怎样用定理 4 求一个线性齐次递推关系的通解形式。

例 7 假设线性齐次递推关系的特征方程的根是 2，2，2，5，5 和 9（即有 3 个根，根 2 的重数为 3，根 5 的重数为 2，根 9 的重数为 1）。那么通解形式是什么？

解 由定理 4，解的一般形式是

$$(\alpha_{1,0} + \alpha_{1,1} n + \alpha_{1,2} n^2) 2^n + (\alpha_{2,0} + \alpha_{2,1} n) 5^n + \alpha_{3,0} 9^n$$ ■

我们现在说明在特征方程有 3 重根时如何用定理 4 求解常系数线性齐次递推关系。

例 8 找出具有初始条件 $a_0 = 1$，$a_1 = -2$，$a_2 = -1$ 的递推关系

$$a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$$

的解。

解 这个递推关系的特征方程是

$$r^3 + 3r^2 + 3r + 1 = 0$$

因为 $r^3 + 3r^2 + 3r + 1 = (r+1)^3$，特征方程只有一个 3 重根 $r = -1$。由定理 4，这个递推关系的解是下述形式

$$a_n = \alpha_{1,0}(-1)^n + \alpha_{1,1} n (-1)^n + \alpha_{1,2} n^2 (-1)^n$$

为求出常数 $\alpha_{1,0}$，$\alpha_{1,1}$，$\alpha_{1,2}$，使用初始条件，得到

$$\begin{aligned} a_0 &= 1 = \alpha_{1,0} \\ a_1 &= -2 = -\alpha_{1,0} - \alpha_{1,1} - \alpha_{1,2} \\ a_2 &= -1 = \alpha_{1,0} + 2\alpha_{1,1} + 4\alpha_{1,2} \end{aligned}$$

这 3 个方程的联立解是 $\alpha_{1,0} = 1$，$\alpha_{1,1} = 3$，$\alpha_{1,2} = -2$。于是，这个递推关系和给定初始条件的唯一解是序列 $\{a_n\}$，其中

$$a_n = (1 + 3n - 2n^2)(-1)^n$$ ■

5.2.3 常系数线性非齐次的递推关系

我们已经知道如何求解常系数线性齐次的递推关系。是否有一种相对简单的技术来求解

如像 $a_n=3a_{n-1}+2n$ 的常系数线性但是非齐次的递推关系呢？我们将看到仅仅对某些特定类的递推关系存在肯定的回答。

递推关系 $a_n=3a_{n-1}+2n$ 是一个常系数线性非齐次递推关系，即形如

$$a_n=c_1a_{n-1}+c_2a_{n-2}+\cdots+c_ka_{n-k}+F(n)$$

的递推关系的例子，其中 $c_1, c_2, \cdots, c_k$ 是实数，$F(n)$是只依赖于 n 且不恒为 0 的函数。递推关系

$$a_n=c_1a_{n-1}+c_2a_{n-2}+\cdots+c_ka_{n-k}$$

叫做相伴的齐次递推关系。它在非齐次递推关系的求解中起了重要的作用。

例 9 递推关系 $a_n=a_{n-1}+2^n$，$a_n=a_{n-1}+a_{n-2}+n^2+n+1$，$a_n=3a_{n-1}+n3^n$ 和$a_n=a_{n-1}+a_{n-2}+a_{n-3}+n!$ 是常系数线性非齐次递推关系。相伴的线性齐次递推关系分别是 $a_n=a_{n-1}$，$a_n=a_{n-1}+a_{n-2}$，$a_n=3a_{n-1}$和 $a_n=a_{n-1}+a_{n-2}+a_{n-3}$。 ■

关于常系数线性非齐次递推关系的一个关键事实是每个解都是一个特解与相伴的线性齐次递推关系的一个解之和，正如下面的定理所述。

定理 5 如果$\{a_n^{(p)}\}$是常系数非齐次线性递推关系

$$a_n=c_1a_{n-1}+c_2a_{n-2}+\cdots+c_ka_{n-k}+F(n)$$

的一个特解，那么每个解都是$\{a_n^{(p)}+a_n^{(h)}\}$的形式，其中$\{a_n^{(h)}\}$是相伴的齐次递推关系

$$a_n=c_1a_{n-1}+c_2a_{n-2}+\cdots+c_ka_{n-k}$$

的一个解。

证 由于$\{a_n^{(p)}\}$是非齐次递推关系的特解，我们知道

$$a_n^{(p)}=c_1a_{n-1}^{(p)}+c_2a_{n-2}^{(p)}+\cdots+c_ka_{n-k}^{(p)}+F(n)$$

现在假设 $\{b_n\}$ 是常系数非齐次递推关系的第二个解，使得

$$b_n=c_1b_{n-1}+c_2b_{n-2}+\cdots+c_kb_{n-k}+F(n)$$

从第二个等式减去第一个等式得

$$b_n-a_n^{(p)}=c_1(b_{n-1}-a_{n-1}^{(p)})+c_2(b_{n-2}-a_{n-2}^{(p)})+\cdots+c_k(b_{n-k}-a_{n-k}^{(p)})$$

从而得到 $\{b_n-a_n^{(p)}\}$ 是相伴的线性齐次递推关系的一个解，比如说是$\{a_n^{(h)}\}$。因此，对所有的 n 有 $b_n=a_n^{(p)}+a_n^{(h)}$。

由定理 5，我们看到求解常系数非齐次递推关系的关键是找一个特解。然后每个解都是这个特解和相伴的齐次递推关系的一个解之和。尽管不存在对每个函数$F(n)$都有效的一般性方法来求这种解，但有某些技术对特定的函数类$F(n)$，例如多项式函数与常数的幂函数有效。例 10 和例 11 就说明了这一点。

例 10 求递推关系 $a_n=3a_{n-1}+2n$ 的所有的解。具有 $a_1=3$ 的解是什么？

解 为求解这个常系数线性非齐次递推关系，我们需要求解它的相伴的线性齐次方程并且找到一个关于给定非齐次方程的特解。相伴的线性齐次方程是 $a_n = 3a_{n-1}$。它的解是 $a_n^{(h)} = \alpha 3^n$，其中 α 是常数。

我们现在找一个特解。因为$F(n) = 2n$ 是 n 的 1 次多项式，解的一个合理的尝试就是 n 的线性函数，比如说 $p_n = cn + d$，其中 c 和 d 是常数。为确定是否存在这种形式的解，假设 $p_n = cn + d$ 是一个这样的解。那么方程 $a_n - 3a_{n-1} + 2n$ 就变成$cn + d = 3(c(n-1) + d) + 2n$。化简和归并同类项得$(2+2c)n + (2d - 3c) = 0$。从而，$cn + d$ 是一个解，当且仅当 $2 + 2c = 0$ 和 $2d - 3c = 0$。这说明 $cn + d$ 是一个解，当且仅当 $c = -1$ 和$d = -3/2$。因而，$a_n^{(p)} = -n - 3/2$ 是一个特解。

根据定理 5 所有的解都是下述形式

$$a_n = a_n^{(p)} + a_n^{(h)} = -n - 3/2 + \alpha \cdot 3^n$$

其中 α 是常数。

为找出具有 $a_1 = 3$ 的解，在得到的通解公式中令 $n = 1$。我们有 $3 = -1 - 3/2 + 3\alpha$，这就推出 $\alpha = 11/6$。要找的解是$a_n = -n - 3/2 + (11/6)3^n$。 ■

例 11 求出下述递推关系

$$a_n = 5a_{n-1} - 6a_{n-2} + 7^n$$

的所有的解。

解 这是一个线性非齐次递推关系。它的相伴的齐次递推关系

$$a_n = 5a_{n-1} - 6a_{n-2}$$

的解是 $a_n^{(h)} = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n$，其中 α_1 和 α_2 是常数。因为 $F(n) = 7^n$，一个合理的解是 $a_n^{(p)} = C \cdot 7^n$，其中 C 是常数。把这些项代入递推关系得 $C \cdot 7^n = 5C \cdot 7^{n-1} - 6C \cdot 7^{n-2} + 7^n$。提出公因式 7^{n-2}，这个等式变成 $49C = 35C - 6C + 49$，从而推出 $20C = 49$ 或 $C = 49/20$。于是，$a_n^{(p)} = (49/20)7^n$ 是特解。由定理 5，所有的解都有下述形式

$$a_n = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n + (49/20)7^n$$ ■

在例 10 和 11 中，我们凭经验猜想了一个特定形式的解。在两种情况下，我们都能找到特解。这并不是偶然的。每当$F(n)$是 n 的多项式和一个常数的 n 次幂之积时，我们就恰好知道一个特解是什么形式，正如定理 6 所述。定理 6 的证明作为一个挑战性的练习留给读者。

定理 6 假设$\{a_n\}$满足线性非齐次递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$$

其中 $c_1, c_2, \cdots, c_k$ 是实数，且

$$F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots b_1 n + b_0) s^n,$$

其中 $b_0, b_1, \cdots, b_t$ 和 s 是实数。当 s 不是相伴的线性齐次递推关系的特征方程的根时，

存在一个下述形式的特解：

$$n^m(p_t n^t + p_{t-1}n^{t-1} + \cdots + p_1 n + p_0)s^n$$

注意当 s 是相伴的线性齐次递推关系的特征方程的 m 重根时，因式 n^m 保证给出的特解不是相伴的线性齐次递推关系的一个解。我们下面给出一个例子说明定理 6 所提供的特解形式。

例 12　当 $F(n)=3^n$, $F(n)=n3^n$, $F(n)=n^2 2^n$ 和 $F(n)=(n^2+1)3^n$ 时，线性非齐次递推关系 $a_n=6a_{n-1}-9a_{n-2}+F(n)$ 的特解有什么形式？

解　相伴的线性齐次递推关系是 $a_n=6a_{n-1}-9a_{n-2}$。它的特征方程 $r^2-6r+9=(r-3)^2=0$ 有一个 2 重的单根。$F(n)$ 的形式为 $P(n)s^n$，其中 $P(n)$ 是一个多项式，s 是一个常数。为应用定理 6，我们需要知道 s 是否是这个特征方程的根。

由于 $s=3$ 是重数 $m=2$ 的根而 $s=2$ 不是根，定理 6 告诉我们如果 $F(n)=3^n$，特解的形式是 $p_0 n^2 3^n$；如果 $F(n)=n3^n$，特解的形式是 $n^2(p_1 n+p_0)3^n$；如果 $F(n)=n^2 2^n$，特解的形式是 $(p_2 n^2+p_1 n+p_0)2^n$；如果 $F(n)=(n^2+1)3^n$，特解的形式是 $n^2(p_2 n^2+p_1 n+p_0)3^n$。

■

在求解定理 6 所谈的那种类型的递推关系时，若 $s=1$ 一定要小心处理。特别是把定理用于 $F(n)=b_t n^t+b_{t-1}n^{t-1}+\cdots+b_1 n+b_0$，参数 s 取值 $s=1$ 时（尽管项 1^n 没有明确地出现）的情况。根据这个定理，解的形式就依赖于是否 1 是相伴的线性齐次递推关系的特征方程的根。这将在例 13 中说明，它说明了怎样用定理 6 找出前 n 个正整数之和的公式。

例 13　设 a_n 是前 n 个正整数的和，即

$$a_n = \sum_{k=1}^{n} k$$

注意，a_n 满足线性非齐次递推关系

$$a_n = a_{n-1} + n$$

（为从前 $n-1$ 个正整数的和 a_{n-1} 得到前 n 个正整数的和 a_n，只需加上 n 即可。）注意初始条件是 $a_1=1$。

对于 a_n 的相伴的线性齐次递推关系是

$$a_n = a_{n-1}$$

这个齐次递推关系的解是 $a_n^{(h)}=c(1)^n=c$，其中 c 是一个常数。为找到 $a_n=a_{n-1}+n$ 的所有的解，我们仅需要找一个特解。由定理 6，由于 $F(n)=n=n\cdot(1)^n$ 且 $s=1$ 是相伴的线性齐次递推关系的特征方程的 1 阶根，存在一个形如 $n(p_1 n+p_0)=p_1 n^2+p_0 n$ 的特解。

把它代入递推关系得到 $p_1 n^2+p_0 n=p_1(n-1)^2+p_0(n-1)+n$。化简得 $n(2p_1-1)+(p_0-p_1)=0$，这意味着 $2p_1-1=0$ 和 $p_0-p_1=0$，即 $p_0=p_1=1/2$。因此

$$a_n^{(p)} = \frac{n^2}{2} + \frac{n}{2} = \frac{n(n+1)}{2}$$

是一个特解。所以，原递推关系 $a_n=a_{n-1}+n$ 的所有的解由 $a_n=a_n^{(h)}+a_n^{(p)}=c+n(n+$

1)/2 给出。由于 $a_1=1$，我们有 $1=a_1=c+1\cdot 2/2=c+1$，故 $c=0$。从而 $a_n=n(n+1)/2$。(这和 1.7 节练习 22 的公式一样。) ■

练习

1. 确定下面哪些是常系数线性齐次递推关系，如果是，找出它们的阶。
   a) $a_n=3a_{n-1}+4a_{n-2}+5a_{n-3}$
   b) $a_n=2na_{n-1}+a_{n-2}$
   c) $a_n=a_{n-1}+a_{n-4}$
   d) $a_n=a_{n-1}+2$
   e) $a_n=a_{n-1}^2+a_{n-2}$
   f) $a_n=a_{n-2}$
   g) $a_n=a_{n-1}+n$
2. 确定下面哪些是常系数线性齐次递推关系，如果是，找出它们的阶。
   a) $a_n=3a_{n-2}$　　b) $a_n=3$
   c) $a_n=a_{n-1}^2$　　d) $a_n=a_{n-1}+2a_{n-3}$
   e) $a_n=a_{n-1}/n$　　f) $a_n=a_{n-1}+a_{n-2}+n+3$
   g) $a_n=4a_{n-2}+5a_{n-4}+9a_{n-7}$
3. 求解下述具有给定初始条件的递推关系。
   a) $a_n=2a_{n-1}$，$n\geqslant 1$，$a_0=3$
   b) $a_n=a_{n-1}$，$n\geqslant 1$，a_0-2
   c) $a_n=5a_{n-1}-6a_{n-2}$，$n\geqslant 2$，$a_0=1$，$a_1=0$
   d) $a_n=4a_{n-1}-4a_{n-2}$，$n\geqslant 2$，$a_0=6$，$a_1=8$
   e) $a_n=-4a_{n-1}-4a_{n-2}$，$n\geqslant 2$，$a_0=0$，$a_1=1$
   f) $a_n=4a_{n-2}$，$n\geqslant 2$，$a_0=0$，$a_1=4$
   g) $a_n=a_{n-2}/4$，$n\geqslant 2$，$a_0=1$，$a_1=0$
4. 求解下述具有给定初始条件的递推关系
   a) $a_n=a_{n-1}+6a_{n-2}$，$n\geqslant 2$，$a_0=3$，a_1-6
   b) $a_n=7a_{n-1}-10a_{n-2}$，$n\geqslant 2$，$a_0=2$，$a_1=1$
   c) $a_n=6a_{n-1}-8a_{n-2}$，$n\geqslant 2$，$a_0=4$，$a_1=10$
   d) $a_n=2a_{n-1}-a_{n-2}$，$n\geqslant 2$，$a_0=4$，$a_1=1$
   e) $a_n=a_{n-2}$，$n\geqslant 2$，$a_0=5$，$a_1=-1$
   f) $a_n=-6a_{n-1}-9a_{n-2}$，$n\geqslant 2$，$a_0=3$，$a_1=-3$
   g) $a_{n+2}=-4a_{n+1}+5a_n$，$n\geqslant 0$，$a_0=2$，$a_1=8$
5. 使用 5.1 节练习 29 描述的两个信号在 n 微秒内可以传送多少不同的信息？
6. 如果传送 1 个信号要 1 微秒，传送另外 2 个信号中的每一个都需要 2 微秒，且在信息中一个信号紧接着下一个信号，使用这 3 个不同的信号在 n 微秒内可以传送多少个不同的信息？
7. 使用 1×2 和 2×2 的块铺满一块 $2\times n$ 的长方形板有多少种方式？
8. 一个关于每年捕捞龙虾数的模型基于如下的假设：1 年捕捞的龙虾数是前 2 年捕捞龙虾数

的平均值。

a) 找出一个关于 $\{L_n\}$ 的递推关系，其中 L_n 是在这个模型的假设下第 n 年捕捞的龙虾数。

b) 如果在第 1 年捕捞了 100 000 只龙虾且第 2 年捕捞了 300 000 只龙虾，求 L_n。

9. 年初把一笔 100 000 美元的钱存入一个投资基金。在每年的最后一天得到两份红利。第一份红利是当年账上钱数的 20%。第二份红利是前一年账上钱数的 45%。

a) 如果不允许取钱，找出一个关于 $\{P_n\}$ 的递推关系，其中 P_n 是第 n 年末账上的钱数。

b) 如果不允许取钱，n 年以后账上有多少钱？

*10. 证明定理 2。

11. 卢卡斯数满足递推关系

$$L_n = L_{n-1} + L_{n-2}$$

和初始条件 $L_0 = 2$ 和 $L_1 = 1$。

a) 证明 $L_n = f_{n-1} + f_{n+1}$，$n = 2, 3, \cdots$，其中 f_n 是第 n 个斐波那契数。

b) 求出卢卡斯数的显示公式。

12. 求解 $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$，$n = 3, 4, 5, \cdots$，且 $a_0 = 3$，$a_1 = 6$，$a_2 = 0$。

13. 求解 $a_n = 7a_{n-2} + 6a_{n-3}$，$a_0 = 9$，$a_1 = 10$，$a_2 = 32$。

14. 求解 $a_n = 5a_{n-2} - 4a_{n-4}$，$a_0 = 3$，$a_1 = 2$，$a_2 = 6$，$a_3 = 8$。

15. 求解 $a_n = 2a_{n-1} + 5a_{n-2} - 6a_{n-3}$，$a_0 = 7$，$a_1 = -4$，$a_2 = 8$。

*16. 证明定理 3。

17. 证明下述涉及斐波那契数和二项式系数的恒等式：

$$f_{n+1} = C(n,0) + C(n-1,1) + \cdots + C(n-k,k)$$

其中 n 是正整数且 $k = \lfloor n/2 \rfloor$。[提示：设 $a_n = C(n,0) + C(n-1,1) + \cdots + C(n-k,k)$。证明序列 $\{a_n\}$ 和斐波那契序列满足的递推关系和初始条件一样。]

18. 求解递推关系 $a_n = 6a_{n-1} - 12a_{n-2} + 8a_{n-3}$，$a_0 = -5$，$a_1 = 4$，$a_2 = 88$。

19. 求解递推关系 $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$，$a_0 = 5$，$a_1 = -9$，$a_2 = 15$。

20. 找出递推关系 $a_n = 8a_{n-2} - 16a_{n-4}$ 的解的一般形式。

21. 如果线性齐次递推关系的特征方程的根是 1，1，1，1，−2，−2，−2，3，3，−4，那么它的解的一般形式是什么？

22. 如果线性齐次递推关系的特征方程的根是 −1，−1，−1，2，2，5，5，7，那么它的解的一般形式是什么？

23. 考虑非齐次线性递推关系 $a_n = 3a_{n-1} + 2^n$。

a) 证明 $a_n = -2^{n+1}$ 是这个递推关系一个的解。

b) 使用定理 5 找出这个递推关系的所有的解。

c) 找出具有 $a_0 = 1$ 的解。

24. 考虑非齐次线性递推关系 $a_n = 2a_{n-1} + 2^n$。

a) 证明 $a_n = n2^n$ 是这个递推关系的一个解。

b) 使用定理 5 找出这个递推关系的所有的解。

c) 找出具有 $a_0 = 2$ 的解。

25. a）确定常数 A 和 B 的值使得 $a_n = An + B$ 是递推关系 $a_n = 2a_{n-1} + n + 5$ 的一个解。
    b）使用定理 5 找出这个递推关系的所有的解。
    c）找出这个递推关系具有 $a_0 = 4$ 的解。
26. 什么是线性非齐次递推关系 $a_n = 6a_{n-1} - 12a_{n-2} + 8a_{n-3} + F(n)$的特解的一般形式？如果
    a）$F(n) = n^2$
    b）$F(n) = 2^n$
    c）$F(n) = n2^n$
    d）$F(n) = (-2)^n$
    e）$F(n) = n^2 2^n$
    f）$F(n) = n^3(-2)^n$
    g）$F(n) = 3$
27. 什么是线性非齐次递推关系$a_n = 8a_{n-2} - 16a_{n-4} + F(n)$的特解的一般形式？如果
    a）$F(n) = n^3$
    b）$F(n) = (-2)^n$
    c）$F(n) = n2^n$
    d）$F(n) = n^2 4^n$
    e）$F(n) = (n^2 - 2)(-2)^n$
    f）$F(n) = n^4 2^n$
    g）$F(n) = 2$
28. a）找出递推关系 $a_n = 2a_{n-1} + 2n^2$ 的所有的解。
    b）找出 a）中的递推关系具有初始条件 $a_1 = 4$ 的解。
29. a）找出递推关系 $a_n = 2a_{n-1} + 3^n$ 的所有的解。
    b）找出 a）的递推关系具有初始条件 $a_1 = 5$ 的解。
30. a）找出递推关系 $a_n = -5a_{n-1} - 6a_{n-2} + 42 \cdot 4^n$ 的所有的解。
    b）找出这个递推关系具有初始条件 $a_1 = 56$ 和 $a_2 = 278$ 的解。
31. 找出递推关系 $a_n = 5a_{n-1} - 6a_{n-2} + 2^n + 3n$ 的所有的解［提示：找形如 $qn2^n + p_1 n + p_2$ 的特解，其中 q，p_1，p_2 是常数。］
32. 找出递推关系 $a_n = 2a_{n-1} + 3 \cdot 2^n$ 的所有的解。
33. 找出递推关系$a_n = 4a_{n-1} - 4a_{n-2} + (n+1)2^n$ 的所有的解。
34. 找出递推关系 $a_n = 7a_{n-1} - 16a_{n-2} + 12a_{n-3} + n4^n$ 的具有 $a_0 = -2$，$a_1 = 0$，$a_2 = 5$ 的解。
35. 找出递推关系 $a_n = 4a_{n-1} - 3a_{n-2} + 2^n + n + 3$ 的具有 $a_0 = 1$，$a_1 = 4$ 的解。
36. 设 a_n 是前 n 个完全平方的和，即 $a_n = \sum_{k=1}^{n} k^2$。证明序列 $\{a_n\}$ 满足线性非齐次递推关系 $a_n = a_{n-1} + n^2$ 和初始条件 $a_1 = 1$。通过使用定理 6 求解这个递推关系确定关于 a_n 的公式。

37. 设 a_n 是前 n 个三角形数的和，即 $a_n=\sum_{k=1}^{n}t_k$，其中 $t_k=k(k+1)/2$。证明 $\{a_n\}$ 满足线性非齐次递推关系 $a_n=a_{n-1}+n(n+1)/2$ 和初始条件 $a_1=1$。通过使用定理 6 求解这个递推关系确定关于 a_n 的公式。

38. a) 求线性齐次递推关系 $a_n=2a_{n-1}-2a_{n-2}$ 的特征根。（注：这些根是复数。）

 b) 求 a) 的递推关系具有 $a_0=1$ 和 $a_1=2$ 的解。

*39. a) 求线性齐次递推关系 $a_n=a_{n-4}$ 的特征根。（注：这些根包含复数。）

 b) 求 a) 的递推关系具有 $a_0=1$，$a_1=0$，$a_2=-1$ 和 $a_3=1$ 的解。

*40. 求解联立递推关系

$$a_n=3a_{n-1}+2b_{n-1}$$
$$b_n=a_{n-1}+2b_{n-1}$$

初始条件 $a_0=1$ 和 $b_0=2$。

*41. a) 用练习 4 中的第 n 个斐波那契数 f_n 的公式证明 f_n 是最接近

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

的整数。

 b) 确定对哪些 n 有 f_n 大于

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

对哪些 n 有 f_n 小于

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

42. 证明如果 $a_n=a_{n-1}+a_{n-2}$，$a_0=s$ 和 $a_1=t$，其中 s 和 t 是常数，那么对所有的正整数 n 有 $a_n=sf_{n-1}+tf_n$。

43. 用斐波那契数的项表示线性非齐次递推关系 $a_n=a_{n-1}+a_{n-2}+1$ 的解，其中 $n\geqslant 2$，$a_0=0$，$a_1=1$ [提示：令 $b_n=a_n+1$ 并对序列 b_n 应用练习 42。]

*44. (要求线性代数）令 A_n 是 $n\times n$ 矩阵，它的主对角线上都是 2，对角线元素旁边的所有位置是 1，其余的全是 0。找一个关于 A_n 的行列式 d_n 的递推关系。求解这个递推关系并找到一个关于 d_n 的公式。

45. 假设留在岛上的每对用遗传工程培育的兔子在一个月大时生出 2 对新兔子，在两个月大和以后的每个月都生出 6 对新兔子。没有兔子死去，也没有兔子从岛上离开。

 a) 一对新生的兔子留在岛上，求出与 n 个月后岛上兔子对数有关的递推关系。

 b) 通过求解 a) 中的递推关系确定一对新生的兔子留在岛上 n 个月以后岛上的兔子对数。

46. 假设初始在一个岛上有 2 只山羊。由于自然繁殖，岛上的山羊数每年加倍，并且每年有些山羊被带来或被带走。

 a) 假定每年另有 100 只山羊被放到岛上，构造一个关于第 n 年初岛上山羊数的递推关系。

 b) 求解 a) 的递推关系来找出第 n 年初岛上的山羊数。

c) 假定对于每个 $n \geq 3$，在第 n 年有 n 只山羊从岛上带走，构造一个关于第 n 年初岛上山羊数的递推关系。

d) 求解 c) 的关于第 n 年初岛上山羊数的递推关系。

47. 在一个充满活力的新软件公司，一个新女雇员的初始工资为 50 000 美元，公司允诺每年底她的工资将是她前一年工资的 2 倍，并且她在公司的每年都将额外增加 10 000 美元。

a) 构造一个与被雇用的第 n 年她的工资数有关的递推关系。

b) 求解这个递推关系找出她被雇用的第 n 年的工资。

某些线性递推关系没有常系数，但也可以被系统的求解。这就是形如 $f(n)a_n = g(n)a_{n-1} + h(n)$ 的递推关系的情况。练习 48～50 说明了这一点。

*48. a) 证明递推关系

$$f(n)a_n = g(n)a_{n-1} + h(n)$$

其中 $n \geq 1$，$a_0 = C$，可以转变成如下形式的递推关系

$$b_n = b_{n-1} + Q(n)h(n)$$

其中 $b_n = g(n+1)Q(n+1)a_n$，满足

$$Q(n) = (f(1)f(2)\cdots f(n-1))/(g(1)g(2)\cdots g(n))$$

b) 使用 a) 求解原来的递推关系以得到

$$a_n = \frac{C + \sum_{i=1}^{n} Q(i)h(i)}{g(n+1)Q(n+1)}$$

*49. 使用练习 48 求解递推关系 $(n+1)a_n = (n+3)a_{n-1} + n, n \geq 1, a_0 = 1$。

50. 可以证明当以随机的顺序排序 n 个元素时，快速排序算法（在 8.4 节练习中描述）所做的平均比较次数满足递推关系

$$C_n = n + 1 + \frac{2}{n}\sum_{k=0}^{n-1} C_k$$

$n = 1, 2, \cdots$，且初始条件 $C_0 = 0$。

a) 证明 $\{C_n\}$ 也满足递推关系 $nC_n = (n+1)C_{n-1} + 2n$，$n = 1, 2, \cdots$。

b) 使用练习 48 求解 a) 的递推关系以找到关于 C_n 的显示公式。

**51. 证明定理 4。

**52. 证明定理 6。

5.3 分而治之关系

5.3.1 引言

许多递归算法把一个给定输入的问题划分成一个或多个小问题。连续施用这种划分直到可以很快地找到这些较小问题的解。例如，在执行一个二分检索时把对一个元素在表中的搜索减少成对该元素在长度减半的表中的搜索。我们继续施用这种分解直到只剩下一个元素。这种递归算法的另一个例子就是整数乘法的过程，它将两个整数相乘的问题分解成三组位数减半的整数相乘。这种分解连续施用直到只剩下一位的整数为止。这些过程叫做*分而治之算法*。这一节将研究在这种算法的复杂性分析中所产生的递推关系。

5.3.2 分而治之关系

假设一个算法把一个规模为 n 的问题分成 a 个子问题，其中每个子问题的规模是 n/b（为简单起见，假设 b 整除 n；实际上，较小的问题的规模常常是小于等于或者大于等于 n/b 的最近的整数）。此外，假设当这个规模为 n 的问题分解成较小的问题时还需要总量为 $g(n)$ 的额外的运算。那么，如果 $f(n)$ 表示求解这个问题所需的运算数，则得出 f 满足递推关系

$$f(n)=af(n/b)+g(n)$$

这就叫做分而治之递推关系。

例 1 在 2.1 节我们引入了二分检索算法。当 n 是偶数时，这个二分检索算法把对某个元素在长度为 n 的搜索序列中的搜索转变成对同一元素在长度 $n/2$ 的搜索序列中的二分检索（因此，规模为 n 的问题已经被分解成规模 $n/2$ 的问题）。为执行这个分解需要 2 次比较（1 次是为了确定要用到表的哪一半，另 1 次是为了确定表是否还有项留下来）。所以，如果 $f(n)$ 是在规模为 n 的搜索序列中搜索一个元素所需要的比较次数，那么当 n 是偶数时 $f(n)=f(n/2)+2$。 ■

例 2 考虑下面的查找序列 a_1，a_2，…，a_n 中最小和最大元素的算法。如果 $n=1$，那么 a_1 就是最大和最小的元素。如果 $n>1$，把这个序列分成两个序列，或者两者有同样多的元素，或者一个集合比另一个集合多一个元素。问题就归约成查找两个较小序列的最大和最小元素。比较两个较小集合的最大和最小元素从而得到全体的最大和最小元素，原问题的解就得到了。

设 $f(n)$ 是找 n 元集的最小和最大元素所需要的总的比较次数。我们已经说明了当 n 是偶数时一个规模为 n 的问题可以归约成两个规模为 $n/2$ 的问题，这里要使用 2 次比较，一次是比较两个集合的最小元素，而另一次是比较两个集合的最大元素。当 n 是偶数时就得到递推关系 $f(n)=2f(n/2)+2$。 ■

例 3 令人惊讶的是存在许多比整数乘法的传统算法（在 2.4 节描述过）更有效的算法。这里描述的一个有效的算法，就用到了分而治之技术。这个快速的乘法算法开始把每个 $2n$ 位的二进制整数分成两块，每块 n 位。然后，原来的 $2n$ 位的二进制整数的乘法被分解成 3 个 n 位二进制数的乘法，再加上移位和加法。

假设 a 和 b 是两个整数的 $2n$ 位的二进制表达式（为了使得它们等长，如果需要的话，可在这些表达式前面加上若干个 0）。令

$$a=(a_{2n-1}a_{2n-2}\cdots a_1a_0)_2,\quad b=(b_{2n-1}b_{2n-2}\cdots b_1b_0)_2$$

令

$$a=2^nA_1+A_0,\quad b=2^nB_1+B_0$$

其中

$$A_1=(a_{2n-1}\cdots a_{n+1}a_n)_2,\quad A_0=(a_{n-1}\cdots a_1a_0)_2$$

$$B_1=(b_{2n-1}\cdots b_{n+1}b_n)_2, B_0=(b_{n-1}\cdots b_1b_0)_2$$

快速整数乘法算法是基于恒等式

$$ab=(2^{2n}+2^n)A_1B_1+2^n(A_1-A_0)(B_0-B_1)+(2^n+1)A_0B_0$$

关于这个恒等式的一个重要的事实就是它证明了两个 $2n$ 位的二进制数的乘法可以用 3 个 n 位二进制数的乘法加上加法、减法以及移位来实现。这证明了如果 $f(n)$ 是两个 n 位二进制数相乘所需的按位运算的总数，那么

$$f(2n)=3f(n)+Cn$$

这个等式的理由如下：3 次 n 位整数的乘法可以使用 $3f(n)$ 次按位运算实现。每次加法、减法和移位使用的运算次数是 n 位运算的常数倍，而 Cn 表示由这些运算用到的总的按位运算数。 ■

例 4 存在一个 $n\times n$ 矩阵相乘的算法，当 n 是偶数时，它使用 7 次 $(n/2)\times(n/2)$ 矩阵的乘法和 15 次 $(n/2)\times(n/2)$ 矩阵的加法。于是，如果 $f(n)$ 是用到的运算数（乘法和加法），那么当 n 是偶数时有

$$f(n)=7f(n/2)+15n^2/4$$ ■

正如例 1～4 所示，在许多不同的情况中都出现了形如 $f(n)=af(n/b)+g(n)$ 的递推关系。可以对满足这种递推关系的函数的阶作出估计。假设 f 满足这个递推关系，其中 n 可被 b 整除。令 $n=b^k$，其中 k 是一个整数。那么

$$\begin{aligned}f(n)&=af(n/b)+g(n)\\&=a^2f(n/b^2)+ag(n/b)+g(n)\\&=a^3f(n/b^3)+a^2g(n/b^2)+ag(n/b)+g(n)\\&\vdots\\&=a^kf(n/b^k)+\sum_{j=0}^{k-1}a^jg(n/b^j)\end{aligned}$$

由于 $n/b^k=1$，从而有

$$f(n)=a^kf(1)+\sum_{j=0}^{k-1}a^jg(n/b^j)$$

我们可以使用这个关于 $f(n)$ 的等式估计满足分而治之关系的函数的阶。

定理 1 设 f 是满足递推关系

$$f(n)=af(n/b)+c$$

的增函数，其中 n 被 b 整除，$a\geqslant 1$，b 是大于 1 的整数，c 是一个正实数，那么

$$f(n)=\begin{cases}O(n^{\log_b a}),\text{如果 } a>1\\O(\log n),\text{如果 } a=1\end{cases}$$

证 首先令 $n=b^k$。由定理前面的讨论得到的关于 $f(n)$ 的表达式和 $g(n)=c$，

$$f(n)=a^kf(1)+\sum_{j=0}^{k-1}a^jc=a^kf(1)+c\sum_{j=0}^{k-1}a^j$$

先考虑 $a=1$ 的情况。那么有

$$f(n)=f(1)+ck$$

由于 $n=b^k$，我们有 $k=\log_b n$。于是

$$f(n)=f(1)+c\log_b n$$

当 n 不是 b 的幂时，对某个正整数 k，有 $b^k<n<b^{k+1}$。由于 f 是增加的，故 $f(n)\leqslant f(b^{k+1})=f(1)+c(k+1)=(f(1)+c)+ck\leqslant(f(1)+c)+c\log_b n$。因此，当 $a=1$ 时在两种情况下 $f(n)$ 都是 $O(\log n)$。

现在假设 $a>1$。首先假定 $n=b^k$，k 是正整数。由几何级数的求和公式（3.2 节例 6）得

$$\begin{aligned}f(n)&=a^kf(1)+c(a^k-1)/(a-1)\\&=a^k[f(1)+c/(a-1)]-c/(a-1)\\&=C_1n^{\log_b a}+C_2\end{aligned}$$

这是由于 $a^k=a^{\log_b n}=n^{\log_b a}$（见附录 1 的练习 4），其中 $C_1=[f(1)+c/(a-1)]$ 和 $C_2=-c/(a-1)$。

现在假设 n 不是 b 的幂。那么 $b^k<n<b^{k+1}$，其中 k 是一个非负整数。由于 f 是增加的，

$$\begin{aligned}f(n)&\leqslant f(b^{k+1})=C_1a^{k+1}+C_2\\&\leqslant(C_1a)a^{\log_b n}+C_2\\&\leqslant(C_1a)n^{\log_b a}+C_2\end{aligned}$$

这是由于 $k\leqslant\log_b n<k+1$。

于是 $f(n)$ 是 $O(n^{\log_b a})$。

注意 这个证明对于 $f(n)$ 给出了一个显示公式，这里 $n=b^k$。

下面的例子说明怎样使用定理 1。

例 5 设 $f(n)=5f(n/2)+3$ 和 $f(1)=7$。求 $f(2^k)$，其中 k 是一个正整数。如果 f 是一个增函数，请估计 $f(n)$。

解 从定理 1 的证明，考虑 $a=5$，$b=2$，$c=3$，我们看到如果 $n=2^k$，那么

$$\begin{aligned}f(n)&=a^k[f(1)+c/(a-1)]+[-c/(a-1)]\\&=5^k[7+(3/4)]-3/4\\&=5^k(31/4)-3/4\end{aligned}$$

又如果 $f(n)$ 是增加的，定理 1 证明了 $f(n)$ 是 $O(n^{\log_b a})=(n^{\log 5})$。■

我们可以使用定理 1 估计二分检索算法和例 2 查找序列的最小和最大元素的算法的计算复杂性。

例 6 估计二分检索使用的比较次数。

解 在例 1 中证明了当 n 是偶数时 $f(n)=f(n/2)+2$，其中 $f(n)$ 是在规模为 n 的序列

实现一个二分检索需要的比较次数。因此得出 $f(n)$ 是 $O(\log n)$。■

例 7　估计用例 2 给定的算法查找序列的最大和最小元素所使用的比较次数。

解　在例 2 我们证明了当 n 是偶数时 $f(n)=2f(n/2)+2$，其中 f 是算法需要的比较次数。于是，由定理 1 得到 $f(n)=O(n^{\log 2})=O(n)$。■

我们现在叙述一个更一般的更复杂的定理，它对于分析分而治之算法的复杂性是非常有用的。

定理 2　设 f 是个满足递推关系

$$f(n)=af(n/b)+cn^d$$

的增函数，其中 $n=b^k$，k 是一个正整数，$a\geqslant 1$，b 是大于 1 的整数，c 和 d 是正实数，那么

$$f(n)=\begin{cases}O(n^d),\text{若 } a<b^d\\ O(n^d\log n),\text{若 } a=b^d\\ O(n^{\log_b a}),\text{若 } a>b^d\end{cases}$$

定理 2 的证明留给读者作为节末的练习 17～21。

例 8　估计使用快速乘法算法做两个 n 位二进制数相乘所需要的按位运算次数。

解　例 3 证明了当 n 是偶数时 $f(n)=3f(n/2)+Cn$，其中 $f(n)$ 是使用快速乘法算法乘两个 n 位二进制数所需要的按位运算次数。于是，由定理 2 得到 $f(n)$ 是 $O(n^{\log 3})$。注意 $\log 3\approx 1.6$。因为通常的乘法算法使用 $O(n^2)$ 次按位运算，所以快速乘法算法对于足够大的整数在时间复杂性方面比通常的算法有了本质的改进。■

例 9　估计使用例 4 的矩阵乘法算法作两个 $n\times n$ 矩阵相乘所需要的乘法和加法次数。

解　令 $f(n)$ 表示做两个 $n\times n$ 矩阵相乘使用例 4 提到的算法所需的加法和乘法次数。当 n 是偶数时我们有 $f(n)=7f(n/2)+15n^2/4$。于是由定理 2 得到 $f(n)$ 是 $O(n^{\log 7})$。注意 $\log 7\approx 2.8$。由于通常的两个 $n\times n$ 矩阵相乘的算法要用 $O(n^3)$ 次加法和乘法，显然，对足够大的整数 n，这个算法比起通常的算法在时间复杂性方面更加有效。■

练习

1. 在 64 个元素的集合中做二分检索需要多少次比较？
2. 在 128 个元素的序列中使用例 2 中的算法查找最大和最小的元素需要多少次比较？
3. 使用快速乘法算法将 $(1\,110)_2$ 与 $(1\,010)_2$ 相乘。
4. 用伪码表示快速乘法算法。
5. 确定在例 3 中的常数 C 的值并且使用它估计用快速乘法算法做两个 64 位二进制数相乘所需要的按位运算次数。
6. 用例 4 引入的算法做两个 32×32 矩阵相乘需要多少次运算？
7. 假设当 n 被 3 整除时有 $f(n)=f(n/3)+1$ 和 $f(1)=1$。求

a) $f(3)$ b) $f(27)$ c) $f(729)$

8. 假设当 n 是偶数时有 $f(n)=2f(n/2)+3$ 和 $f(1)=5$。求

a) $f(2)$ b) $f(8)$ c) $f(64)$ d) $f(1\,024)$

9. 假设当 n 被 5 整除时有 $f(n)=f(n/5)+3n^2$ 和 $f(1)=4$。求

a) $f(5)$ b) $f(125)$ c) $f(3\,125)$

10. 当 $n=2^k$ 时求 $f(n)$，其中 f 满足递推关系 $f(n)=f(n/2)+1, f(1)=1$。

11. 如果 f 是一个增函数，估计练习 10 中的 f 的规模。

12. 当 $n=3^k$ 时求 $f(n)$，其中 f 满足递推关系 $f(n)=2f(n/3)+4, f(1)=1$。

13. 如果 f 是一个增函数，估计练习 12 中的 f 的规模。

14. 假设在一个淘汰锦标赛中有 $n=2^k$ 个队，其中在第一轮有 $n/2$ 场比赛，$n/2=2^{k-1}$ 个赢的队进入第二轮比赛，依此进行。建立一个关于锦标赛的轮数的递推关系。

15. 在练习 14 的淘汰锦标赛中如果有 32 个队，需要进行多少轮比赛？

16. 求解练习 14 所描述的关于锦标赛轮数的递推关系。

在练习 17～21，假设 f 是一个满足递推关系 $f(n)=af(n/b)+cn^d$ 的增函数，$a\geq 1$，b 是大于 1 的整数，c 和 d 是正实数。这些练习提供一个关于定理 2 的证明。

*17. 证明如果 $a=b^d$ 且 n 是 b 的幂，那么 $f(n)=f(1)n^d+cn^d\log_b n$。

18. 使用练习 17 证明如果 $a=b^d$，那么 $f(n)$ 是 $O(n^d\log n)$。

*19. 证明如果 $a\neq b^d$，n 是 b 的幂，那么 $f(n)=C_1n^d+C_2n^{\log_b a}$，其中 $C_1=b^dc/(b^d-a)$ 且 $C_2=f(1)+b^dc/(a-b^d)$。

20. 使用练习 19 证明如果 $a<b^d$，那么 $f(n)$ 是 $O(n^d)$。

21. 使用练习 19 证明如果 $a>b^d$，那么 $f(n)$ 是 $O(n^{\log_b a})$

22. 当 $n=4^k$，求 $f(n)$，其中 f 满足递推关系 $f(n)=5f(n/4)+6n, f(1)=1$。

23. 如果 f 是增函数，估计练习 22 中 f 的规模。

24. 当 $n=2^k$，求 $f(n)$，其中 f 满足递推关系 $f(n)=8f(n/2)+n^2, f(1)=1$。

25. 如果 f 是增函数，估计练习 24 中 f 的规模。

5.4 生成函数

5.4.1 引言

表示序列的一种有效方法就是生成函数，它把序列的项作为一个形式幂级数中变量 x 的幂的系数。可以用生成函数求解许多类型的计数问题，例如在各种限制下选取或分配不同种类物体的方式数，使用不同面额的硬币换一美元的方式数等。也可以用生成函数求解递推关系。它先把关于序列的项的递推关系转换成涉及生成函数的方程，然后求解这个方程并找出关于这个生成函数的封闭形式。从这个封闭形式可以找到生成函数的幂级数的系数，从而求解原来的递推关系。生成函数也可以利用函数之间相对简单的关系来证明组合恒等式，由于这些关系可以转换成涉及序列的项的恒等式。生成函数是有用的工具，除了本节描述的以外，可以用它来研究序列的许多性质，例如建立关于序列的项的渐进公式。

我们从序列的生成函数的定义开始。

定义1 实数序列 a_0，a_1，…，a_k，…的生成函数是无穷级数

$$G(x) = a_0 + a_1 x + \cdots + a_k x^k + \cdots = \sum_{k=0}^{\infty} a_k x^k$$

注意 定义1给出的 $\{a_k\}$ 的生成函数有时叫做 $\{a_k\}$ 的普通生成函数，以和这个序列的其他类型的生成函数相区别。

例1 序列 $\{a_k\}$ 具有 $a_k = 3$，$a_k = k + 1$ 和 $a_k = 2^k$ 的生成函数分别是$\sum_{k=0}^{\infty} 3x^k$，$\sum_{k=0}^{\infty}(k+1)x^k$，$\sum_{k=0}^{\infty} 2^k x^k$。 ■

我们通过置 $a_{n+1} = 0$，$a_{n+2} = 0$，依此下去，把一个有限序列 a_0，a_1，…，a_n 扩充成一个无限序列，就可以定义一个实数的有限序列的生成函数。这个无限序列 $\{a_n\}$ 的生成函数$G(x)$是一个 n 次多项式，因为当 $j > n$ 时没有形如$a_j x^j$ 的项出现，即

$$G(x) = a_0 + a_1 x + \cdots + a_n x^n$$

例2 序列1，1，1，1，1，1的生成函数是什么？

解 1，1，1，1，1，1的生成函数是

$$1 + x + x^2 + x^3 + x^4 + x^5$$

由3.2节的例6有

$$(x^6 - 1)/(x - 1) = 1 + x + x^2 + x^3 + x^4 + x^5$$

因此$G(x) = (x^6 - 1)/(x - 1)$是序列1，1，1，1，1，1的生成函数。 ■

例3 设 m 是正整数。令$a_k = C(m, k), k = 0, 1, 2, \cdots, m$。那么序列 a_0，a_1，…，a_m 的生成函数是什么？

解 这个序列的生成函数是

$$G(x) = C(m,0) + C(m,1)x + C(m,2)x^2 + \cdots + C(m,m)x^m$$

二项式定理证明 $G(x) = (1 + x)^m$。 ■

5.4.2 关于幂级数的有用的事实

当用生成函数求解计数问题时，通常将它们考虑成形式幂级数。这里忽略了这些级数的收敛问题。但是为了应用某些微积分的结果，考虑幂级数对什么 x 收敛有时是很重要的。在我们的讨论中将不涉及收敛性问题。熟悉微积分的读者为了解所涉及级数的收敛性的细节可以参阅有关这方面内容的教科书。

现在我们将叙述某些与无穷级数有关的重要事实，这些将在研究生成函数时用到。这些事实的讨论和相关的结果都可以在微积分教科书中找到。

例4 函数$f(x) = 1/(1 - x)$是序列1，1，1，1，…的生成函数，因为对$|x| < 1$有

$$1/(1 - x) = 1 + x + x^2 + \cdots$$ ■

例5 函数$f(x) = 1/(1 - ax)$是序列1，a，a^2，a^3，…的生成函数，因为当$|ax| < 1$

或等价地说对 $|x|<1/|a|$，$a\neq 0$ 有

$$1/(1-ax)=1+ax+a^2x^2+\cdots$$ ■

我们也需要了解两个生成函数是怎样相加和怎样相乘的。这些结果的证明也可以在微积分教科书中找到。

定理 1 令 $f(x)=\sum_{k=0}^{\infty}a_kx^k, g(x)=\sum_{k=0}^{\infty}b_kx^k$，那么

$$f(x)+g(x)=\sum_{k=0}^{\infty}(a_k+b_k)x^k, f(x)g(x)=\sum_{k=0}^{\infty}\left(\sum_{j=0}^{k}a_jb_{k-j}\right)x^k$$

我们将用下面的例子说明定理 1 的使用。

例 6 设 $f(x)=1/(1-x)^2$。用例 4 求出表达式 $f(x)=\sum_{k=0}^{\infty}a_kx^k$ 中的系数 a_0，a_1，a_2，…。

解 由例 4 看出

$$1/(1-x)=1+x+x^2+x^3+\cdots$$

因此，由定理 1，有

$$1/(1-x)^2=\sum_{k=0}^{\infty}\left(\sum_{j=0}^{k}1\right)x^k=\sum_{k=0}^{\infty}(k+1)x^k$$ ■

注意 这一结果也可以通过微分从例 4 导出。从已知生成函数的恒等式产生新的恒等式的一种有用的技术就是取微商。

为了用生成函数求解许多重要的计数问题，需要在指数不是正整数的情况下应用二项式定理。在叙述推广的二项式定理之前，需要定义推广的二项式系数。

定义 2 设 u 是实数且 k 是非负整数，那么推广的二项式系数定义为

$$\binom{u}{k}=\begin{cases}u(u-1)\cdots(u-k+1)/k!, & 若\ k>0\\ 1, & 若\ k=0\end{cases}$$

例 7 求推广的二项式系数 $\binom{-2}{3}$ 和 $\binom{1/2}{3}$ 的值。

解 在定义 2 中取 $u=-2$ 和 $k=3$ 得

$$\binom{-2}{3}=\frac{(-2)(-3)(-4)}{3!}=-4$$

类似地取 $u=1/2$ 和 $k=3$ 得

$$\begin{aligned}\binom{1/2}{3}&=\frac{(1/2)(1/2-1)(1/2-2)}{3!}\\&=(1/2)(-1/2)(-3/2)/6\\&=1/16\end{aligned}$$ ■

当上边的参数是负整数时，下面的例子对推广的二项式系数提供了一个有用的公式。在

后面的讨论中会用到它。

例 8 当上边的参数是负整数时，推广的二项式系数可以用普通二项式系数的项表示。为此只需注意到

$$\begin{aligned}\binom{-n}{k} &= \frac{(-n)(-n-1)\cdots(-n-r+1)}{r!}\\ &= \frac{(-1)^r n(n+1)\cdots(n+r-1)}{r!}\\ &= \frac{(-1)^r(n+r-1)(n+r-2)\cdots n}{r!}\\ &= \frac{(-1)^r(n+r-1)!}{r!\ (n-1)!}\\ &= (-1)^r\binom{n+r-1}{r}\\ &= (-1)^r C(n+r-1,r)\end{aligned}$$

■

我们现在叙述推广的二项式定理。

定理 2 推广的二项式定理。设 x 是满足 $|x|<1$ 的实数，u 是实数。那么有

$$(1+x)^u = \sum_{k=0}^{\infty}\binom{u}{k}x^k$$

可以用麦克劳林级数的理论证明定理 2。将这个证明留给熟悉这部分微积分内容的读者。

注意 当 u 是正整数时，推广的二项式定理变成 4.3 节给出的二项式定理，因为如果 $k>u$ 有 $\binom{u}{k}=0$。

当指数是负整数时下面的例子说明了定理 2 的使用。

例 9 当 n 是正整数时使用推广的二项式定理求 $(1+x)^{-n}$ 和 $(1-x)^{-n}$ 的生成函数。

解 由推广的二项式定理得

$$(1+x)^{-n} = \sum_{k=0}^{\infty}\binom{-n}{k}x^k$$

使用例 8 所提供的关于 $\binom{-n}{k}$ 的简单公式得到

$$(1+x)^{-n} = \sum_{k=0}^{\infty}(-1)^k C(n+k-1,k)x^k$$

用 $-x$ 代替 x 得到

$$(1-x)^{-n} = \sum_{k=0}^{\infty}C(n+k-1,k)x^k$$

■

表 5-1 归纳了一些经常出现的有用的生成函数。

表 5-1　有用的生成函数

$G(x)$	a_k
$(1+x)^n = \sum_{k=0}^{n} C(n,k)x^k = 1 + C(n,1)x + C(n,2)x^2 + \cdots + x^n$	$C(n,k)$
$(1+ax)^n = \sum_{k=0}^{n} C(n,k)a^k x^k = 1 + C(n,1)ax + C(n,2)a^2x^2 + \cdots + a^n x^n$	$C(n,k)a^k$
$(1+x^r)^n = \sum_{k=0}^{n} C(n,k)x^{rk} = 1 + C(n,1)x^r + C(n,2)x^{2r} + \cdots + x^{rn}$	当 $r \mid k$ 时为 $C(n,k/r)$；否则为0
$\frac{1-x^{n+1}}{1-x} = \sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n$	当 $k \leqslant n$ 时为1；否则为0
$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \cdots$	1
$\frac{1}{1-ax} = \sum_{k=0}^{\infty} a^k x^k = 1 + ax + a^2x^2 + \cdots$	a^k
$\frac{1}{1-x^r} = \sum_{k=0}^{\infty} x^{rk} = 1 + x^r + x^{2r} + \cdots$	当 $r \mid k$ 时为1；否则为0
$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty} (k+1)x^k = 1 + 2x + 3x^2 + \cdots$	$k+1$
$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)x^k = 1 + C(n,1)x + C(n+1,2)x^2 + \cdots$	$C(n+k-1,k) = C(n+k-1,n-1)$
$\frac{1}{(1+x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)(-1)^k x^k = 1 - C(n,1)x + C(n+1,2)x^2 - C(n+2,3)x^2 + \cdots$	$(-1)^k C(n+k-1,k) = (-1)^k C(n+k-1,n-1)$
$\frac{1}{(1-ax)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)a^k x^k = 1 + C(n,1)ax + C(n+1,2)a^2x^2 + \cdots$	$C(n+k-1,k)a^k = C(n+k-1,n-1)a^k$
$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^2}{3!} + \cdots$	$1/k!$
$\ln(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^{k+1}}{k} x^k = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$	$(-1)^{k+1}/k$

注：在大多数微积分书的幂级数部分可以找到关于最后两个生成函数的级数。

5.4.3 计数问题与生成函数

生成函数可以用于求解各种计数问题。特别地，它们可以用于计数各种类型的组合数。在第 4 章我们开发了一些技术计数 n 元素集合的允许重复的 r-组合数，在这些组合中可能存在某些附加的约束。这种问题是与计数形如

$$e_1 + e_2 + \cdots + e_n = C$$

方程的解是等价的，其中 C 是常数，每个 e_i 是可能具有某些约束的非负整数。也可以用生成函数求解这种类型的计数问题，正如下面的例子所说明的。

例 10 求

$$e_1 + e_2 + e_3 = 17$$

的解的个数，其中 e_1，e_2，e_3 是非负整数，满足 $2 \leqslant e_1 \leqslant 5$，$3 \leqslant e_2 \leqslant 6$，$4 \leqslant e_3 \leqslant 7$。

解 具有上述限制的解的个数是

$$(x^2 + x^3 + x^4 + x^5)(x^3 + x^4 + x^5 + x^6)(x^4 + x^5 + x^6 + x^7)$$

的展开式中 x^{17}的系数。这是因为我们在乘积中得到等于 x^{17}的项是通过在第一个和中取项 x^{e_1}，在第二个和中取项 x^{e_2}，在第三个和中取项 x^{e_3}，其中幂指数 e_1，e_2 和 e_3 满足方程 $e_1 + e_2 + e_3 = 17$ 和给定的限制。

不难看出在这个乘积中的 x^{17}的系数是 3。因此，存在 3 个解。（注意计算这个系数与枚举方程的具有给定约束的所有解几乎要做同样多的工作。但是，正如将要看到的，这里说明的方法常常可以用于求解各种各样的具有特殊规则的计数问题。此外，可以用计算机代数系统做这种计算。） ■

例 11 把 8 块相同的饼干分给 3 个不同的孩子，如果每个孩子至少接受 2 块饼干并且不超过 4 块饼干，那么有多少种不同的方式？

解 因为每个孩子至少接受 2 块饼干且不超过 4 块饼干，在关于序列 $\{c_n\}$ 的生成函数中对每个孩子存在一个等于

$$(x^2 + x^3 + x^4)$$

的因式，其中 c_n 是分配 n 块饼干的方式数。因为存在 3 个孩子，这个生成函数是

$$(x^2 + x^3 + x^4)^3$$

我们要求这个乘积中的 x^8 的系数。理由就是在展开式中的 x^8 的项对应于选 3 项的方式数，其中每个因式选 1 项且指数加起来等于 8。此外，来自第一、第二和第三个因式的项的指数分别是第一、第二和第三个孩子接受的饼干数。通过计算说明这个系数等于 6。于是存在 6 种方式分配饼干使得每个孩子至少接受 2 块但是不超过 4 块饼干。 ■

例 12 把价值 1 美元、2 美元和 5 美元的代币插入售货机为价值 r 美元的某种物品付款，使用生成函数确定在代币插入是有序的和无序的两种情况下付款的方式数。（例如为一种价值 3 美元的物品付款，当不管代币插入的次序时存在 2 种方式：插入 3 个 1 美元的代币

或 1 个 1 美元和 1 个 2 美元的代币。当考虑代币插入的次序时有 3 种方式：插入 3 个 1 美元的代币，插入 1 个 1 美元代币然后 1 个 2 美元的代币，插入 1 个 2 美元代币然后 1 个 1 美元代币。)

解　在不考虑代币插入次序的情况下，我们所关心的就是为产生 r 美元的总数所使用的每种代币的数目。因为我们可以使用任意多个 1 美元的代币，任意多个 2 美元的代币，和任意多个 5 美元的代币，答案就是在生成函数

$$(1+x+x^2+x^3+\cdots)(1+x^2+x^4+x^6+\cdots)(1+x^5+x^{10}+x^{15}+\cdots)$$

中的 x^r 的系数。(这个乘积中的第一个因式表示所使用的 1 美元代币，第二个表示所使用的 2 美元代币，第三个表示所使用的 5 美元代币。) 例如，用 1 美元、2 美元和 5 美元为一个价值 7 美元的物品付款的方式数由展开式中 x^7 的系数给出，结果等于 6。

当考虑代币插入的次序时，插入恰好 n 个代币产生总数 r 美元的方式数是

$$(x+x^2+x^5)^n$$

中的 x^r 的系数，因为这 n 个代币中的每一个可能是 1 美元代币、2 美元代币或 5 美元代币。又由于可以插入的代币不限数量，因此当考虑代币插入的次序时，插入恰好 n 个代币产生总数 r 美元的方式数是

$$1+(x+x^2+x^5)+(x+x^2+x^5)^2+\cdots=\frac{1}{1-(x+x^2+x^5)}$$
$$=\frac{1}{1-x-x^2-x^5}$$

中 x^r 的系数，这里我们把插入 0 个代币、1 个代币、2 个代币、3 个代币等等的方式数加起来，同时我们使用了恒等式 $1/(1-x)=1+x+x^2+\cdots$ 且用 $x+x^2+x^5$ 代替 x。例如，用 1 美元、2 美元和 5 美元的代币为一个价值 7 美元的物品付款，当考虑使用代币的次序时，方式数是这个展开式中 x^7 的系数，等于 26 [为看到这个系数等于 26，要把 $(x+x^2+x^5)^k$ 的展开式中 x^7 的系数加起来，其中 $2\leqslant k\leqslant 7$。这项工作可以用大量的手工计算完成，也可以使用一个计算机代数系统。] ■

下面的例子说明了当求解带不同假设的问题时生成函数具有的多功能性。

例 13　假设已经建立了二项式定理，使用生成函数找出 n 元素集合的 k-组合数。

解　集合的 n 个元素中的每一个元素都对生成函数 $f(x)=\sum_{k=0}^{n}a_kx^k$ 贡献了项 $(1+x)$。因此 $f(x)$ 是关于 $\{a_k\}$ 的生成函数，其中 a_k 表示 n 元素集合的 k-组合数。于是，

$$f(x)=(1+x)^n$$

但是由二项式定理，我们有

$$f(x)=\sum_{k=0}^{n}\binom{n}{k}x^k$$

其中

$$\binom{n}{k} = \frac{n!}{k!\ (n-k)!}$$

于是，$C(n,k)$，n 元素集合的 k-组合数是 ■

$$\frac{n!}{k!\ (n-k)!}$$

注意　在 4.3 节使用了关于 n 元素集合的 r-组合数的公式证明了二项式定理。这些例子说明也可以用数学归纳法证明二项式定理，再用二项式定理推导关于 n 元素集合的 r-组合数公式。

例 14　使用生成函数找出当元素允许重复时 n 元素集合的 r-组合数公式。

解　设 $G(x)$ 是关于序列 $\{a_r\}$ 的生成函数，其中 a_r 等于 n 元素集合的允许重复的 r-组合数，即 $G(x)=\sum_{r=0}^{\infty}a_rx^r$。当我们构成允许重复的 r-组合时对 n 元素集合的元素的选择不受限制，这 n 个元素中的每一个元素都对 $G(x)$ 的乘积展开式贡献了 $(1+x+x^2+x^3+\cdots)$ 这个因式。这是由于当一个 r-组合被构成时（要选总共 r 个元素）每个元素都可以被选择 0 次、1 次、2 次、3 次，等等。因为集合中存在 n 个元素，且每一个都对 $G(x)$ 贡献了相同的因式，从而有

$$G(x)=(1+x+x^2+x^3+\cdots)^n$$

只要 $|x|<1$，就有 $1+x+x^2+x^3+\cdots=1/(1-x)$，所以

$$G(x)=1/(1-x)^n=(1-x)^{-n}$$

使用推广的二项式定理，得到

$$(1-x)^{-n}=(1+(-x))^{-n}=\sum_{r=0}^{\infty}\binom{-n}{r}(-x)^r$$

当 r 是正整数时，n 元素集合的允许重复元素的 r-组合数就是这个和式中的 x^r 的系数。因此，使用例 8 我们求出 a_r 等于

$$\begin{aligned}\binom{-n}{r}(-1)^r &= (-1)^rC(n+r-1,r)\cdot(-1)^r\\ &= C(n+r-1,r)\end{aligned}$$ ■

注意这与我们在 4.6 节定理 2 所叙述的结果一样。

例 15　使用生成函数求出从 n 类不同的物体中选择 r 个物体并且每类物体至少选 1 个的方式数。

解　因为我们需要每类物体至少选 1 个，这 n 个类中的每类物体都对序列 $\{a_r\}$ 的生成函数 $G(x)$ 贡献了因式 $(1+x+x^2+x^3+\cdots)$，其中 a_r 是从 n 类不同的物体中选择 r 个物体并且每类物体至少选 1 个的方式数。因此，

$$G(x)=(x+x^2+x^3+\cdots)^n=x^n(1+x+x^2+x^3+\cdots)^n=x^n/(1-x)^n$$

使用推广的二项式定理和例 8，有

$$G(x)=x^n/(1-x)^n$$

$$
\begin{aligned}
&= x^n \cdot (1-x)^{-n} \\
&= x^n \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r \\
&= x^n \sum_{r=0}^{\infty} (-1)^r C(n+r-1,r)(-1)^r x^r \\
&= \sum_{r=0}^{\infty} C(n+r-1,r) x^{n+r} \\
&= \sum_{t=n}^{\infty} C(t-1,t-n) x^t \\
&= \sum_{r=n}^{\infty} C(r-1,r-n) x^r
\end{aligned}
$$

在倒数第二个等式我们令 $t=n+r$ 以使得当 $r=0$ 时 $t=n$ 且 $n+r-1=t-1$，从而对求和进行了移位，然后在最后的等式中用 r 替换 t 作为求和的下标而回到了初始的记号。因此，从 n 类不同的物体中选择 r 个物体，如果每类物体必须至少选 1 个时，存在 $C(r-1,r-n)$ 种方式。■

5.4.4 使用生成函数求解递推关系

我们可以通过找相关的生成函数的显式公式来求解关于一个递推关系和初始条件的解。这可以用下面的例子来说明。

例 16　求解递推关系 $a_k=3a_{k-1}$，$k=1, 2, 3, \cdots$且初始条件 $a_0=2$。

解　设$G(x)$是序列$\{a_k\}$的生成函数，即$G(x)=\sum_{k=0}^{\infty}a_kx^k$。首先注意到

$$
xG(x)=\sum_{k=0}^{\infty}a_kx^{k+1}=\sum_{k=1}^{\infty}a_{k-1}x^k
$$

使用递推关系有

$$
\begin{aligned}
G(x)-3xG(x) &= \sum_{k=0}^{\infty}a_kx^k-3\sum_{k=1}^{\infty}a_{k-1}x^k \\
&= a_0+\sum_{k=1}^{\infty}(a_k-3a_{k-1})x^k \\
&= 2
\end{aligned}
$$

因为 $a_0=2$ 且 $a_k=3a_{k-1}$。于是

$$
G(x)-3xG(x)=(1-3x)G(x)=2
$$

求解$G(x)$得 $G(x)=2/(1-3x)$。使用表 5-1 中的恒等式$1/(1-ax)=\sum_{k=0}^{\infty}a^kx^k$，我们有

$$
G(x)=2\sum_{k=0}^{\infty}3^kx^k=\sum_{k=0}^{\infty}2\cdot 3^kx^k
$$

于是，$a_k=2\cdot 3^k$。■

例 17　设一个有效的编码字是一个包含偶数个 0 的 n 位十进制数字串。令 a_n 表示 n 位

有效编码字的个数。在 5.1 节的例 7 中我们证明了序列 $\{a_n\}$ 满足递推关系

$$a_n = 8a_{n-1} + 10^{n-1}$$

与初始条件 $a_1 = 9$。使用生成函数找出关于 a_n 的显式公式。

解 为了简化关于生成函数的推导，我们通过置 $a_0 = 1$ 将序列扩充，当把这个值赋给 a_0 并且使用递推关系就得到 $a_1 = 8a_0 + 10^0 = 8 + 1 = 9$，这与我们的初始条件一致。（由于存在一个长为 0 的编码字——空串，这也是有意义的。）

用 x^n 乘递推关系的两边得

$$a_n x^n = 8a_{n-1}x^n + 10^{n-1}x^n$$

设 $G(x) = \sum_{k=0}^{\infty} a_k x^k$ 是序列 a_0，a_1，a_2，…的生成函数，从 1 开始对上面的等式两边求和，得到

$$\begin{aligned}
G(x) - 1 &= \sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty}(8a_{n-1}x^n + 10^{n-1}x^n) \\
&= 8\sum_{n=1}^{\infty} a_{n-1}x^n + \sum_{n=1}^{\infty} 10^{n-1}x^n \\
&= 8x\sum_{n=1}^{\infty} a_{n-1}x^{n-1} + x\sum_{n=1}^{\infty} 10^{n-1}x^{n-1} \\
&= 8x\sum_{n=0}^{\infty} a_n x^n + x\sum_{n=0}^{\infty} 10^n x^n \\
&= 8xG(x) + x/(1-10x)
\end{aligned}$$

其中已经使用了例 5 对第二个和进行求值。因此有

$$G(x) - 1 = 8xG(x) + x/(1-10x)。$$

求解 $G(x)$ 得

$$G(x) = \frac{1-9x}{(1-8x)(1-10x)}$$

把等式的右边展开成部分分式（正如在微积分中研究有理函数的积分时所做的）得到

$$G(x) = \frac{1}{2}\left(\frac{1}{1-8x} + \frac{1}{1-10x}\right)$$

两次使用例 5（一次设 $a = 8$，一次设 $a = 10$）得

$$\begin{aligned}
G(x) &= \frac{1}{2}\left[\sum_{n=0}^{\infty} 8^n x^n + \sum_{n=0}^{\infty} 10^n x^n\right] \\
&= \sum_{n=0}^{\infty} \frac{1}{2}(8^n + 10^n)x^n
\end{aligned}$$

于是，证明了

$$a_n = \frac{1}{2}(8^n + 10^n)$$

■

5.4.5 使用生成函数证明恒等式

在第4章我们已经看到怎样使用组合证明方法来建立组合恒等式。这里将显示这种恒等式，还有关于推广的二项式系数的恒等式，都可以使用生成函数来证明，有时候生成函数的方法比其他的方法更简单，特别是用生成函数的封闭形式比使用序列本身更能简化证明过程。我们用下面的例子说明怎样用生成函数证明恒等式。

例 18 使用生成函数证明

$$\sum_{k=0}^{n} C(n,k)^2 = C(2n,n)$$

其中 n 是正整数。

解 首先注意到根据二项式定理$C(2n,n)$是 $(1+x)^{2n}$中x^n 的系数。而我们也有

$$\begin{aligned}(1+x)^{2n} &= [(1+x)^n]^2 \\ &= [C(n,0)+C(n,1)x+C(n,2)x^2+\cdots+C(n,n)x^n]^2\end{aligned}$$

在这个展开式中 x^n 的系数是$C(n,0)C(n,n)+C(n,1)C(n,n-1)+C(n,2)C(n,n-2)+\cdots+C(n,n)C(n,0)$。因为 $C(n,n-k)=C(n,k)$，它等于$\sum_{k=0}^{n}C(n,k)^2$。由于 $C(2n,n)$和$\sum_{k=0}^{n}C(n,k)^2$ 都表示$(1+x)^{2n}$中x^n 的系数，它们一定是相等的。 ■

节末的练习 42 和 43 要求用生成函数来证明帕斯卡恒等式和范德蒙恒等式。

练习

1. 求关于有穷序列 2，2，2，2，2，2 的生成函数。
2. 求关于有穷序列 1，4，16，64，256 的生成函数。
3. 求关于下面每个序列生成函数的封闭形式。(用最明显的选择设定每个序列的通项形式。)
   a) 0，2，2，2，2，2，2，0，0，0，0，0，…
   b) 0，0，0，1，1，1，1，1，1，…
   c) 0，1，0，0，1，0，0，1，0，0，1，…
   d) 2，4，8，16，32，64，128，256，…
   e) $\binom{7}{0}$，$\binom{7}{1}$，$\binom{7}{2}$，…，$\binom{7}{7}$，0，0，0，0，0，…
   f) 2，−2，2，−2，2，−2，2，−2，…
   g) 1，1，0，1，1，1，1，1，1，1，…
   h) 0，0，0，1，2，3，4，…
4. 求关于下面每个序列生成函数的封闭形式。(用最明显的选择设定每个序列的通项形式。)
   a) −1，−1，−1，−1，−1，−1，−1，0，0，0，0，0，0，…
   b) 1，3，9，27，81，243，729，…
   c) 0，0，3，−3，3，−3，3，−3，…
   d) 1，2，1，1，1，1，1，1，1，…
   e) $\binom{7}{0}$，$2\binom{7}{1}$，$2^2\binom{7}{2}$，…，$2^7\binom{7}{7}$，0，0，0，0，…

f) −3，3，−3，3，−3，3，…

g) 0，1，−2，4，−8，16，−32，64，…

h) 1，0，1，0，1，0，1，0，…

5. 求关于序列 $\{a_n\}$ 的生成函数的封闭形式，其中

a) $a_n=5$，对所有的 $n=0, 1, 2, \cdots$

b) $a_n=3^n$，对所有的 $n=0, 1, 2, \cdots$

c) $a_n=2$，对 $n=3, 4, 5, \cdots$ 且 $a_0=a_1=a_2=0$

d) $a_n=2n+3$，对所有的 $n=0, 1, 2, \cdots$

e) $a_n=\binom{8}{n}$，对所有的 $n=0, 1, 2, \cdots$

f) $a_n=\binom{n+4}{n}$，对所有的 $n=0, 1, 2, \cdots$

6. 求关于序列 $\{a_n\}$ 的生成函数的封闭形式，其中

a) $a_n=-1$，对所有的 $n=0, 1, 2, \cdots$

b) $a_n=2^n$，对 $n=1, 2, 3, 4, \cdots$ 且 $a_0=0$

c) $a_n=n-1$，对 $n=0, 1, 2, \cdots$

d) $a_n=1/(n+1)!$，对 $n=0,1,2,\cdots$

e) $a_n=\binom{n}{2}$，对 $n=0, 1, 2, \cdots$

f) $a_n=\binom{10}{n+1}$，对 $n=0, 1, 2, \cdots$

7. 对于下面每一个生成函数给出关于它所确定序列的封闭公式。

a) $(3x-4)^3$

b) $(x^3+1)^3$

c) $1/(1-5x)$

d) $x^3/(1+3x)$

e) $x^2+3x+7+(1/(1-x^2))$

f) $(x^4/(1-x^4))-x^3-x^2-x-1$

g) $x^2/(1-x)^2$

h) $2e^{2x}$

8. 对于下面每一个生成函数给出关于它所确定序列的封闭公式。

a) $(x^2+1)^3$

b) $(3x-1)^3$

c) $1/(1-2x^2)$

d) $x^2/(1-x)^3$

e) $x-1+(1/(1-3x))$

f) $(1+x^3)/(1+x)^3$

*g) $x/(1+x+x^2)$

h) $e^{3x^2}-1$

9. 求出下面每个函数的幂级数中 x^{10} 的系数。

a) $(1+x^5+x^{10}+x^{15}+\cdots)^3$

b) $(x^3+x^4+x^5+x^6+x^7+\cdots)^3$

c) $(x^4+x^5+x^6)(x^3+x^4+x^5+x^6+x^7)(1+x+x^2+x^3+x^4+\cdots)$

d) $(x^2+x^4+x^6+x^8+\cdots)(x^3+x^6+x^9+\cdots)(x^4+x^8+x^{12}+\cdots)$

e) $(1+x^2+x^4+x^6+x^8+\cdots)(1+x^4+x^8+x^{12}+\cdots)(1+x^6+x^{12}+x^{18}+\cdots)$

10. 求出下面每个函数的幂级数中 x^9 的系数。

 a) $(1+x^3+x^6+x^9+\cdots)^3$

 b) $(x^2+x^3+x^4+x^5+x^6+\cdots)^3$

 c) $(x^3+x^5+x^6)(x^3+x^4)(x+x^2+x^3+x^4+\cdots)$

 d) $(x+x^4+x^7+x^{10}+\cdots)(x^2+x^4+x^6+x^8+\cdots)$

 e) $(1+x+x^2)^3$

11. 求出下面每个函数的幂级数中 x^{10}的系数。

 a) $1/(1-2x)$

 b) $1/(1+x)^2$

 c) $1/(1-x)^3$

 d) $1/(1+2x)^4$

 e) $x^4/(1-3x)^3$

12. 求出下面每个函数的幂级数中 x^{12}的系数。

 a) $1/(1+3x)$

 b) $1/(1-2x)^2$

 c) $1/(1+x)^8$

 d) $1/(1-4x)^3$

 e) $x^3/(1+4x)^2$

13. 把 10 个相同的球分给 4 个孩子，如果每个孩子至少得到 2 个球，使用生成函数确定不同的分法数。

14. 把 12 个相同的剧情图片分给 5 个孩子，使得每个孩子至多得到 3 张，使用生成函数确定不同的分法数。

15. 把 15 个相同的动物玩具分给 6 个孩子，使得每个孩子至少得到 1 个但不超过 3 个，使用生成函数确定不同的分法数。

16. 从 3 类百吉饼——鸡蛋的、椒盐的和普通的选 12 个，如果每类至少选 2 个但椒盐的不超过 3 个，使用生成函数确定选法数。

17. 把 25 个相同的多纳圈分给 4 个警官，使得每个警官至少得到 3 个但不超过 7 个，有多少种方式?

18. 从包含 100 个红球、100 个蓝球和 100 个绿球的罐子选 14 个球，使得蓝球不少于 3 个且不多于 10 个。假定不考虑选球的次序，使用生成函数求出选法数。

19. 求序列 $\{c_k\}$ 的生成函数，其中 c_k 是使用 1 美元、2 美元、5 美元和 10 美元纸币换 k 美元的方法数。

20. 求序列 $\{c_k\}$ 的生成函数，其中 c_k 是使用 10 比索、20 比索、50 比索和 100 比索换 k 比索的方法数。

21. 对 $(1+x+x^2+x^3+\cdots)^3$ 展开式中 x^4 的系数给出组合解释。使用这个解释求出这个数。
22. 对 $(1+x+x^2+x^3+\cdots)^n$ 展开式中 x^6 的系数给出组合解释。使用这个解释求出这个数。
23. a) 什么是关于 $\{a_k\}$ 的生成函数？这里的 a_k 是 $x_1+x_2+x_3=k$ 的解的个数，其中 x_1，x_2 和 x_3 是满足 $x_1\geqslant 2$，$0\leqslant x_2\leqslant 3$，$2\leqslant x_3\leqslant 5$ 的整数。
    b) 使用 a) 的答案求 a_6。
24. a) 什么是关于 $\{a_k\}$ 的生成函数？这里的 a_k 是 $x_1+x_2+x_3-x_4=k$ 的解的个数，其中 x_1，x_2，x_3 和 x_4 是满足 $x_1\geqslant 3$，$1\leqslant x_2\leqslant 5$，$0\leqslant x_3\leqslant 4$，$x_4\geqslant 1$ 的整数。
    b) 使用 a) 的答案求 a_7。
25. 解释怎样使用生成函数找到用 3 分、4 分和 20 分的邮票在信封上贴满 r 分邮费的方式数。
    a) 假设不考虑贴邮票的次序。
    b) 假设邮票贴成一行并且考虑贴的次序。
    c) 当不考虑贴邮票的次序时，使用 a) 的答案确定用 3 分、4 分和 20 分的邮票在信封上贴满 46 分邮费的方式数。（建议使用计算机代数程序。）
    d) 当考虑贴邮票的次序时，使用 b) 的答案确定用 3 分、4 分和 20 分的邮票在信封上贴满一行 46 分邮费的方式数。（建议使用计算机代数程序。）
26. a) 重复掷一个骰子，考虑掷的次序并且使得掷出的点数之和为 n，证明关于这种方式数的生成函数是 $1/(1-x-x^2-x^3-x^4-x^5-x^6)$。
    b) 使用 a) 求出重复掷一个骰子、考虑掷的次序并且使得掷出的总点数为 8 的方式数。（建议使用计算机代数程序。）
27. 使用生成函数（如果需要的话，也使用计算机代数程序）求出换 1 美元的方式数。
    a) 用 10 美分和 25 美分。
    b) 用 5 美分、10 美分和 25 美分。
    c) 用 1 美分、10 美分和 25 美分。
    d) 用 1 美分、5 美分、10 美分和 25 美分。
28. 使用生成函数（如果需要的话，也使用计算机代数程序）求出用 1 美分、5 美分、10 美分和 25 美分换 1 美元的方式数，使得
    a) 1 美分不超过 10 个。
    b) 1 美分不超过 10 个且 5 美分不超过 10 个。
    c) 硬币不超过 10 个。
29. 使用生成函数求出换 100 美元的方式数。
    a) 用 10 美元、20 美元和 50 美元纸币。
    b) 用 5 美元、10 美元、20 美元和 50 美元纸币。
    c) 用 5 美元、10 美元、20 美元和 50 美元纸币，并且每种纸币至少使用 1 张。
    d) 用 5 美元、10 美元和 20 美元纸币，并且每种纸币至少使用 1 张但不超过 4 张。
30. 如果 $G(x)$ 是关于序列 $\{a_k\}$ 的生成函数，那么关于下述每个序列的生成函数是什么？

a) $2a_0$，$2a_1$，$2a_2$，$2a_3$，…

b) 0，a_0，a_1，a_2，a_3，…

c) 0，0，0，0，a_2，a_3，…

d) a_2，a_3，a_4，…

e) a_1，$2a_2$，$3a_3$，$4a_4$，… [提示：这里需要微积分。]

f) a_0^2，$2a_0a_1$，$a_1^2+2a_0a_2$，$2a_0a_3+2a_1a_2$，$2a_0a_4+2a_1a_3+a_2^2$，…

31. 如果 $G(x)$ 是关于序列 $\{a_k\}$ 的生成函数，那么关于下述每个序列的生成函数是什么？

a) 0，0，0，a_3，a_4，a_5，…

b) a_0，0，a_1，0，a_2，0，…

c) 0，0，0，0，a_0，a_1，a_2，…

d) a_0，$2a_1$，$4a_2$，$8a_3$，$16a_4$，…

e) a_0，$a_1/2$，$a_2/3$，$a_3/4$，… [提示：这里需要微积分。]

f) a_0，a_0+a_1，$a_0+a_1+a_2$，$a_0+a_1+a_2+a_3$，…

32. 使用生成函数求解递推关系 $a_k=7a_{k-1}$，初始条件 $a_0=5$。

33. 使用生成函数求解递推关系 $a_k=3a_{k-1}+2$，初始条件 $a_0=1$。

34. 使用生成函数求解递推关系 $a_k=3a_{k-1}+4^{k-1}$，初始条件 $a_0=1$。

35. 使用生成函数求解递推关系 $a_k=5a_{k-1}-6a_{k-2}$，初始条件 $a_0=6$ 和 $a_1=30$。

36. 使用生成函数求解递推关系 $a_k=a_{k-1}+2a_{k-2}+2^k$，初始条件 $a_0=4$ 和 $a_1=12$。

37. 使用生成函数求解递推关系 $a_k=4a_{k-1}-4a_{k-2}+k^2$，初始条件 $a_0=2$ 和 $a_1=5$。

38. 使用生成函数求解递推关系 $a_k=2a_{k-1}+3a_{k-2}+4^k+6$，初始条件 $a_0=20$ 和 $a_1=60$。

39. 使用生成函数找出关于斐波那契数的显式公式。

*40. a) 证明如果 n 是正整数，那么

$$\binom{-1/2}{n}=\frac{\binom{2n}{n}}{(-4n)^n}$$

b) 使用推广的二项式定理和 a) 证明对于一切非负整数 n 在 $(1-4x)^{-1/2}$ 的展开式中 x^n 的系数是 $\binom{2n}{n}$。

*41. (需要微积分) 设 $\{C_n\}$ 是卡特朗数，即具有初值 $C_0=C_1=1$ 的递推关系 $C_n=\sum_{k=0}^{n-1}C_kC_{n-1-k}$ 的解（见 5.1 节例 8）。

a) 证明如果 $G(x)$ 是关于卡特朗数的序列的生成函数，那么 $xG(x)^2-G(x)+1=0$。(使用初始条件) 断定 $G(x)=(1-\sqrt{1-4x})/(2x)$。

b) 使用练习 40 断定

$$G(x)=\sum_{n=0}^{\infty}\frac{1}{n+1}\binom{2n}{n}x^n$$

从而

$$C_n=\frac{1}{n+1}\binom{2n}{n}$$

42. 当 n 和 r 是 $r<n$ 的正整数，使用生成函数证明帕斯卡恒等式：$C(n,r)=C(n-1,r)+C(n-1,r-1)$。[提示：使用恒等式 $(1+x)^n=(1+x)^{n-1}+x(1+x)^{n-1}$。]

43. 使用生成函数证明范德蒙恒等式：$C(m+n,r)=\sum_{k=0}^{r}C(m,r-k)C(n,k)$，其中 m，n 和 r 是非负整数，且 r 不超过 m 或 n。[提示：看在 $(1+x)^{m+n}=(1+x)^m(1+x)^n$ 两边的 x^r 的系数。]

44. 这个练习说明了怎样使用生成函数推导前 n 个平方数之和的公式。

 a) 证明 $(x^2+x)/(1-x)^4$ 是关于序列 $\{a_n\}$ 的生成函数，其中 $a_n=1^2+2^2+\cdots+n^2$。

 b) 使用 a) 找出关于和 $1^2+2^2+\cdots+n^2$ 的显示公式。

关于序列 $\{a_n\}$ 的指数生成函数是级数

$$\sum_{n=0}^{\infty}\frac{a_n}{n!}x^n$$

例如，关于序列 1，1，1，… 的指数生成函数是 $\sum_{n=0}^{\infty}x^n/n!=e^x$（你将发现这个级数在下面的练习中很有用。）注意 e^x 是关于序列 1，1，1/2!，1/3!，1/4!，… 的普通生成函数。

45. 求一个关于序列 $\{a_n\}$ 的指数生成函数的封闭形式，其中

 a) $a_n=2$ b) $a_n=(-1)^n$

 c) $a_n=3^n$ d) $a_n=n+1$

 e) $a_n=1/(n+1)$

46. 求一个关于序列 $\{a_n\}$ 的指数生成函数的封闭形式，其中

 a) $a_n=(-2)^n$ b) $a_n=-1$ c) $a_n=n$

 d) $a_n=n(n-1)$ e) $a_n=1/((n+1)(n+2))$

47. 求以下述函数为指数生成函数的序列。

 a) $f(x)=e^{-x}$

 b) $f(x)=3e^{2x}$

 c) $f(x)=e^{3x}-3e^{2x}$

 d) $f(x)=(1-x)+e^{-2x}$

 e) $f(x)=e^{-2x}-(1/(1-x))$

 f) $f(x)=e^{-3x}-(1+x)+(1/(1-2x))$

 g) $f(x)=e^{x^2}$

48. 求以下述函数为指数生成函数的序列。

 a) $f(x)=e^{3x}$ b) $f(x)=2e^{-3x+1}$

 c) $f(x)=e^{4x}+e^{-4x}$ d) $f(x)=(1+2x)+e^{3x}$

 e) $f(x)=e^x-(1/(1+x))$

 f) $f(x)=xe^x$ g) $f(x)=e^{x^3}$

49. 一个编码系统用 8 进制（基为 8）数字串对信息编码。一个编码字是有效的，当且仅当它包含偶数个 7。

a) 求一个关于 n 位长有效编码字个数的线性非齐次递推关系。初始条件是什么？

b) 使用5.2节的定理6解这个递推关系。

c) 用生成函数解这个递推关系。

*50. 一个编码系统用4进制数字串（即数字来自集合{0，1，2，3}）对信息编码。一个编码字是有效的，当且仅当它包含偶数个0和偶数个1。设 a_n 等于长为 n 的有效编码字个数。此外令 b_n 为具有偶数个0和奇数个1的 n 位4进制数字串个数，c_n 为具有奇数个0和偶数个1的 n 位4进制数字串个数，d_n 为具有奇数个0和奇数个1的 n 位4进制数字串个数。

a) 证明 $d_n=4^n-a_n-b_n-c_n$。使用这个式子证明 $a_{n+1}=2a_n+b_n+c_n$，$b_{n+1}=b_n-c_n+4^n$ 和 $c_{n+1}=c_n-b_n+4^n$。

b) a_1，b_1，c_1 和 d_1 是什么？

c) 使用a) 和b) 求出 a_3，b_3，c_3 和 d_3。

d) 使用a) 的递推关系和b) 的初始条件分别建立与序列 $\{a_n\}$，$\{b_n\}$ 和 $\{c_n\}$ 的生成函数 $A(x)$，$B(x)$ 和 $C(x)$ 相关的三个方程。

e) 求解d) 的方程得到关于 $A(x)$，$B(x)$ 和 $C(x)$ 的显示公式并且利用这些公式得到关于 a_n，b_n，c_n 和 d_n 的显示公式。

在研究整数 n 的不同类型的剖分个数时生成函数是很有用的。一个正整数的剖分是把这个整数写成正整数之和，和中的整数允许重复并且不考虑次序。例如，5的剖分（不加限制）是1+1+1+1+1，1+1+1+2，1+1+3，1+2+2，1+4，2+3和5。练习51~56说明了这种应用。

51. 证明在 $1/((1-x)(1-x^2)(1-x^3)\cdots)$ 的形式幂级数展开式中 x^n 的系数 $p(n)$ 等于 n 的剖分数。

52. 证明在 $1/((1-x)(1-x^3)(1-x^5)\cdots)$ 的形式幂级数展开式中 x^n 的系数 $p_o(n)$ 等于 n 剖分成奇整数的方式数，即把 n 写成正奇数之和的方式数，其中不管这些奇数的次序并且允许重复。

53. 证明在 $(1+x)(1+x^2)(1+x^3)\cdots$ 的形式幂级数展开式中 x^n 的系数 $p_d(n)$ 等于 n 剖分成不相等的整数的方式数，即把 n 写成正整数之和的方式数，其中不管这些整数的次序并且不允许重复。

54. 对于 $1\leqslant n\leqslant 8$，通过对每个整数写出每一个不同类型的剖分求 $p_o(n)$ 和 $p_d(n)$，其中 $p_o(n)$ 是 n 剖分成允许重复的奇整数的方式数，$p_d(n)$ 是 n 剖分成不相等的整数的方式数。

55. 证明如果 n 是正整数，那么 n 剖分成不相等的整数的方式数等于 n 剖分成允许重复的奇整数的方式数；即 $p_o(n)=p_d(n)$。[提示：证明关于 $p_o(n)$ 和 $p_d(n)$ 的生成函数相等。]

**56. (需要微积分) 使用关于 $p(n)$ 的生成函数证明对某个常数 C，$p(n)\leqslant e^{C\sqrt{n}}$。[Hardy和Ramanujan证明了 $p(n)\approx e^{\pi\sqrt{2/3}\sqrt{n}}/(4\sqrt{3}n)$，这意味着当 n 达到无限时 $p(n)$ 与右边的比达到1。]

假定 X 是样本空间 S 的随机变量，使得 $X(s)$ 对于所有的 $s\in S$ 是非负整数。X 的概率

生成函数是

$$G_x(X)=\sum_{k=0}^{\infty}P(X(s))=k)x^k$$

57.（需要微积分）证明如果 G_x 是随机变量 X 的概率生成函数，使得 $X(s)$ 对于所有的 $s\in S$ 是非负整数，那么

a) $G_X(1)=1$　　b) $E(X)=G_X'(1)$

c) $V(X)=G_X''(1)+G_X'(1)-G_X'(1)^2$

58. 作独立的伯努利实验，每次实验成功的概率为 p。设 X 是随机变量，且如果第 n 次实验出现首次成功，X 的值就是 n。

a) 求关于概率生成函数 G_x 的封闭公式。

b) 使用练习 57 和 a）中得到的关于概率生成函数的封闭公式求 X 的期望值和方差。

59. 设 m 是正整数，当作独立的伯努利实验时每次实验成功的概率为 p。设 X_m 是随机变量，且如果第 $(n+m)$ 次实验出现第 m 次成功则 X_m 的值就是 n。

a) 使用第 4 章的补充练习 44 证明概率生成函数 G_{x_m} 由 $G_{x_m}(x)=p^m/(1-qx)^m$ 给出，其中 $q=1-p$。

b) 使用练习 57 和 a）中得到的关于概率生成函数的封闭公式求 X_m 的期望值和方差。

60. 证明如果 X 和 Y 是样本空间 S 上的独立的随机变量，$X(s)$ 和 $Y(s)$ 对于所有 $s\in S$ 为非负整数，那么 $G_{X+Y}(x)=G_X(x)G_Y(x)$。

5.5　容斥

5.5.1　引言

一个离散数学班包含 30 个女生和 50 个二年级学生。在这个班里有多少个女生或二年级学生？如果没有更多的信息，这个问题是没法求解的。把女生数和二年级学生数加起来不一定能得出正确的结果，因为二年级的女生可能被计数了两次。这个事实说明在班里的女生或二年级学生数是班里的女生数与二年级学生数之和减去二年级的女生数。在 4.1 节曾经引入过求解这种计数问题的技术。这里我们将把在那一节引入的思想加以推广，以求解更为广泛的计数问题。

5.5.2　容斥原理

两个有穷集的并集中存在多少个元素？在 1.5 节中证明了两个集合 A 和 B 的并集中的元素数是这些集合的元素数之和减去其交集中的元素数，即

$$|A\cup B|=|A|+|B|-|A\cap B|$$

正如我们在 4.1 节证明的，这个关于两个集合并集中元素数的公式在计数问题中是很有用的。下面的例子进一步说明了这个公式的用处。

例 1　一个离散数学班包含 25 个计算机科学专业的学生，13 个数学专业的学生和 8 个同时主修数学和计算机科学两个专业的学生。如果每个学生主修数学专业、计算机科学专业，或者同时主修这两个专业，那么班里有多少个学生？

解　设 A 是这个班里计算机科学专业的学生的集合，B 是这个班里数学专业的学生的集合。那么 $A\cap B$ 是班里主修数学和计算机科学两个专业的学生的集合。因为这个班的每个学生或者主修计算机科学，或者主修数学（或者同时主修两个专业），从而得到这个班里的学生数是 $|A\cup B|$。于是

$$\begin{aligned}|A\cup B| &= |A|+|B|-|A\cap B|\\ &= 25+13-8\\ &= 30\end{aligned}$$

因此，这个班有 30 个学生。这个计算在图 5-5 中说明。■

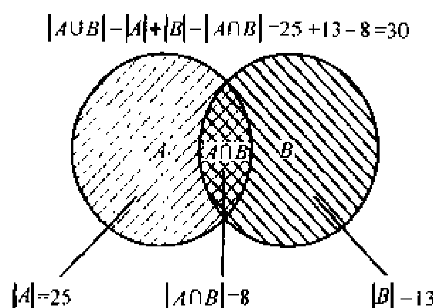


图 5-5　离散数学班的学生的集合

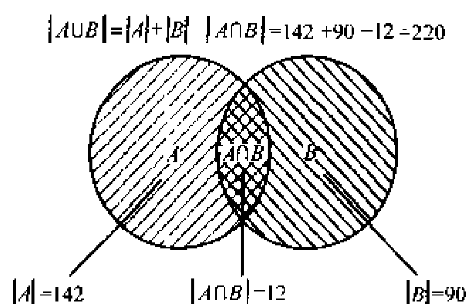


图 5-6　不超过 1 000 的可被 7 或 11 整除的正整数的集合

例 2　有多少个不超过 1 000 的正整数可以被 7 或 11 整除？

解　设 A 是不超过 1 000 且可被 7 整除的正整数的集合，B 是不超过 1 000 且可被 11 整除的正整数的集合，那么 $A\cup B$ 是不超过 1 000 且可被 7 或 11 整除的正整数的集合，且 $A\cap B$ 是不超过 1 000 且可被 7 和 11 同时整除的正整数的集合。由 2.3 节的例 2，我们知道在不超过 1 000 的正整数中有 $\lfloor 1\,000/7\rfloor$ 个整数可被 7 整除，并且有 $\lfloor 1\,000/11\rfloor$ 个整数可被 11 整除。由于 7 和 11 是互素的，被 7 和 11 同时整除的整数就是被 $7\cdot 11$ 整除的整数。因此，有 $\lfloor 1000/(7\cdot 11)\rfloor$ 个不超过 1 000 的正整数可被 7 和 11 同时整除。于是存在

$$\begin{aligned}|A\cup B| &= |A|+|B|-|A\cap B|\\ &= \left\lfloor\frac{1\,000}{7}\right\rfloor+\left\lfloor\frac{1\,000}{11}\right\rfloor+\left\lfloor\frac{1\,000}{7\cdot 11}\right\rfloor\\ &= 142+90-12\\ &= 220\end{aligned}$$

个正整数不超过 1 000 且可被 7 或 11 整除。这个计算在图 5-6 中说明。■

下面的例子说明怎样求有穷全集中在两个集合的并集之外的元素数。

例 3　假设你们学校有 1 807 个新生。这些学生中有 453 人选了一门计算机科学课，567 人选了一门数学课，299 人同时选了计算机科学课和数学课。有多少学生既没有选计算机科学课也没有选数学课？

解　为找出既没有选数学课也没有选计算机科学课的新生数，就要从新生总数中减去选

了其中一门课的学生数。设 A 是选了一门计算机课的所有新生的集合，且 B 是选了一门数学课的所有新生的集合。于是 $|A|=453$，$|B|=567$，且 $|A\cap B|=299$。选了一门计算机科学或数学课的学生数是

$$|A\cup B|=|A|+|B|-|A\cap B|=453+567-299=721$$

因此，存在 $1\ 807-721=1\ 086$ 个新生既没选计算机科学课也没选数学课。这个计算在图 5-7中说明。■

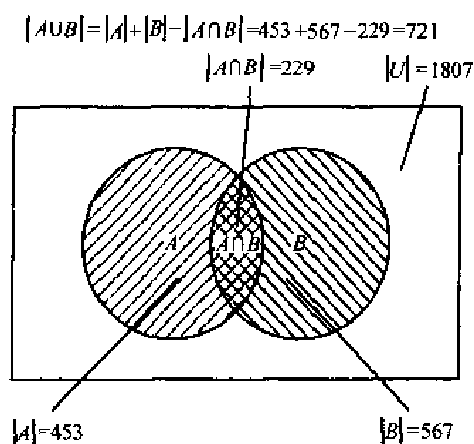


图 5-7 既没选计算机科学课也没选数学课的新生的集合

在本节的后面将说明怎样求有限个集合的并集中的元素数。这个结果叫作容斥原理。设 n 是任意正整数，在考虑 n 个集合的并集之前，先推导与 3 个集合 A，B，C 的并集中的元素数有关的公式。为推导这个公式，首先注意到以下事实：$|A|+|B|+|C|$ 对 3 个集合中那些恰好在其中 1 个集合的元素只计数了 1 次，恰好在其中 2 个集合的元素计数了 2 次，恰好在其中 3 个集合的元素计数了 3 次。这个结果在图 5-8 a) 中说明。

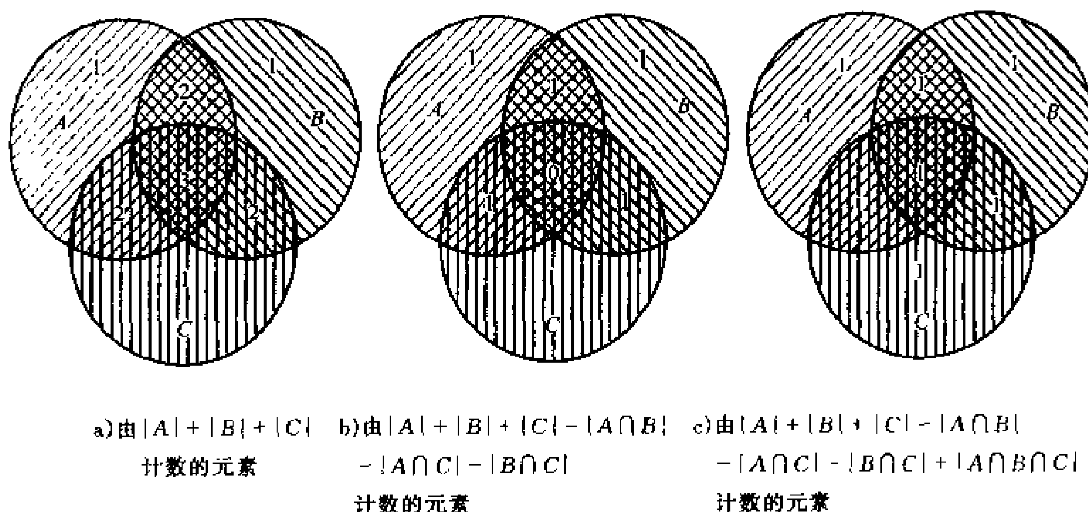


图 5-8 求关于 3 个集合的并集中元素数的公式

为了去掉在多个集合中元素的重复计数，减去这3个集合中的每2个集合的交集中的元素数，得到

$$|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|$$

这个表达式对恰好出现在其中1个集合的元素仍旧计数1次。恰好出现在其中2个集合的元素也被它计数1次，因为2个集合的交集有3个，而这种元素只出现在其中之一。但是，那些出现在3个集合的元素将被这个表达式计数0次，因为它们将会出现在所有的两两相交的3个交集中。这个结果显示在图5-8b）中。

为了纠正这个漏计，还要加上3个集合交集中的元素数。这个最后的表达式对每个元素计数了1次，不管它是在1个、2个还是在3个集合里。于是

$$|A\cup B\cup C|=|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|$$

这个公式显示在图5-8 c）中。

下面的例子说明了怎样使用这个公式。

例4 1 232个学生选了西班牙语课，879个学生选了法语课，114个学生选了俄语课。103个学生选了西班牙语和法语课，23个学生选了西班牙语和俄语课，14个学生选了法语和俄语课。如果2092个学生至少在西班牙语、法语和俄语课中选1门，有多少个学生选了所有这3门语言课？

解 设 S 是选西班牙语课的学生集合，F 是选法语课的学生集合，R 是选俄语课的学生集合。那么

$$|S|=1\,232 \qquad |F|=879 \qquad |R|=114$$
$$|S\cap F|=103 \qquad |S\cap R|=23 \qquad |F\cap R|=14$$

且

$$|S\cup F\cup R|=2092$$

把这些量代入等式

$$|S\cup F\cup R|=|S|+|F|+|R|-|S\cap F|-|S\cap R|-|F\cap R|+|S\cap F\cap R|$$

得

$$2\,092=1\,232+879+114-103-23-14+|S\cap F\cap R|$$

求解 $|S\cap F\cap R|$ 得出 $|S\cap F\cap R|=7$。因此存在7个学生选了西班牙语、法语和俄语课。这个结果在图5-9中说明。 ■

我们现在将叙述和证明容斥原理，它将告诉我们在有限个有穷集的并集中有多少个元素。

定理1 容斥原理。设 A_1，A_2，…，A_n 是有穷集。那么

$$|A_1\cup A_2\cup\cdots A_n|=\sum_{1\le i\le n}|A_i|-\sum_{1\le i<j\le n}|A_i\cap A_j|$$
$$+\sum_{1\le i<j<k\le n}|A_i\cap A_j\cap A_k|-\cdots+(-1)^{n+1}|A_1\cap A_2\cap\cdots\cap A_n|$$

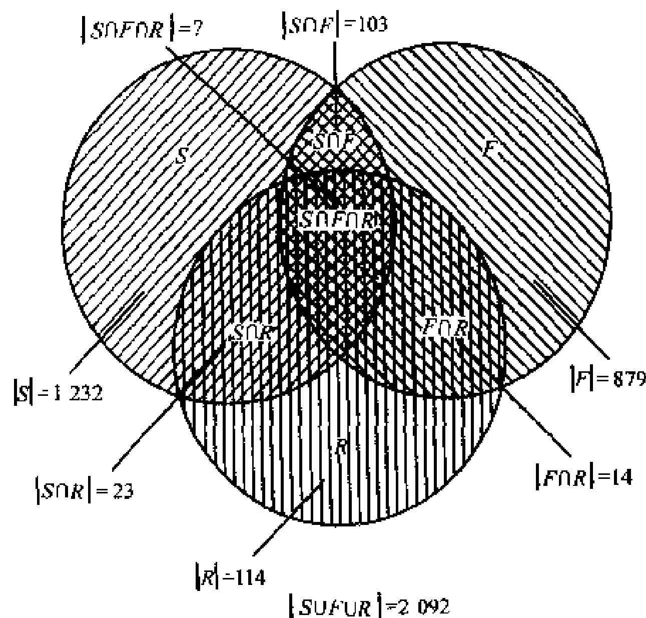


图 5-9 选了西班牙语、法语和俄语课程的学生集合

证 我们将通过证明并集中的每个元素在等式右边恰好被计数 1 次来证明这个公式。假设 a 恰好是 A_1，A_2，…，A_n 中 r 个集合的成员，其中 $1\leqslant r\leqslant n$。这个元素被 $\sum|A_i|$ 计数了 $C(r,1)$ 次，被 $\sum|A_i\cap A_j|$ 计数了 $C(r,2)$ 次。一般说来，它被涉及 m 个 A_i 集合的求和计数了 $C(r,m)$ 次。于是，这个元素恰好被等式右边的表达式计数了

$$C(r,1)-C(r,2)+C(r,3)-\cdots+(-1)^{r+1}C(r,r)$$

次。我们的目标是求出这个值。由 4.3 节的定理 7，我们有

$$C(r,0)-C(r,1)+C(r,2)-\cdots+(-1)^{r}C(r,r)=0$$

于是

$$1=C(r,0)=C(r,1)-C(r,2)+C(r,3)-\cdots+(-1)^{r+1}C(r,r)$$

因此，并集中的每个元素在等式右边的表达式中恰好被计数 1 次。这就证明了容斥原理。

对于每个正整数 n，容斥原理对于 n 个集合并集的元素数给出了一个公式。对于 n 个集合的集合族的每一个非空子集的交，在这个公式中都存在一项计数了它的元素。因此在这个公式中有 2^n-1 项。 □

例 5 对于 4 个集合的并集中的元素数给出一个公式。

解 容斥原理显示

$$\begin{aligned}|A_1\cup A_2\cup A_3\cup A_4| &= |A_1|+|A_2|+|A_3|+|A_4|\\ &-|A_1\cap A_2|-|A_1\cap A_3|-|A_1\cap A_4|-|A_2\cap A_3|-|A_2\cap A_4|-|A_3\cap A_4|\\ &+|A_1\cap A_2\cap A_3|+|A_1\cap A_2\cap A_4|+|A_1\cap A_3\cap A_4|+|A_2\cap A_3\cap A_4|\\ &-|A_1\cap A_2\cap A_3\cap A_4|.\end{aligned}$$

注意这个公式包含 15 个不同的项，对于 $\{A_1, A_2, A_3, A_4\}$ 的每个非空子集有一项。 ■

练习

1. 在 $A_1 \cup A_2$ 中存在多少个元素？如果在 A_1 中存在 12 个元素，在 A_2 中存在 18 个元素，并且
   a) $A_1 \cap A_2 = \varnothing$
   b) $|A_1 \cap A_2| = 1$
   c) $|A_1 \cap A_2| = 6$
   d) $A_1 \subseteq A_2$
2. 一个学院有 345 个学生选了微积分课，212 个学生选了离散数学课，188 个学生同时选了微积分和离散数学课。有多少学生选了微积分或离散数学课？
3. 一项调查显示，在美国 96% 的家庭至少有 1 台电视机，98% 的家庭有电话，95% 的家庭有电话且至少有 1 台电视机。在美国有百分之几的家庭既没有电话也没有电视机？
4. 一个关于个人计算机的市场报告说 650 000 个拥有计算机的人将在下一年为自己的机器买调制解调器，并且 1 250 000 人将至少买 1 个软件包。如果这个报告说 1 450 000 个人将买调制解调器或至少买 1 个软件包，那么有多少人将买调制解调器并且至少买 1 个软件包？
5. 求 $A_1 \cup A_2 \cup A_3$ 中的元素数，如果每个集合有 100 个元素，并且
   a) 这些集合是两两不交的。
   b) 每对集合中存在 50 个公共元素，并且没有元素在所有这 3 个集合里。
   c) 每对集合中存在 50 个公共元素，并且有 25 元素在所有这 3 个集合里。
   d) 这些集合是相等的。
6. 求 $A_1 \cup A_2 \cup A_3$ 中的元素数，如果 A_1 中有 100 个元素，A_2 中有 1 000 个元素，A_3 中有 10 000 个元素，并且
   a) $A_1 \subseteq A_2$ 且 $A_2 \subseteq A_3$。
   b) 这些集合是两两不交的。
   c) 在每对集合中存在 2 个公共元素，并且没有元素在所有这 3 个集合里。
7. 一个学校有 2 504 个计算机科学专业的学生，其中 1 876 人选修了 Pascal，999 人选修了 Fortran，345 人选修了 C，876 人选修了 Pascal 和 Fortran，231 人选修了 Fortran 和 C，290 人选修了 Pascal 和 C。如果 189 个学生选了 Fortran，Pascal 和 C，那么 2 504 个学生中有多少学生没有选这 3 门程序设计语言课中的任何一种？
8. 一项关于 270 个大学生的调查显示 64 人喜欢芽甘蓝，94 人喜欢花椰菜，58 人喜欢花椰菜，26 人喜欢芽甘蓝和椰菜，28 人喜欢芽甘蓝和花椰菜，22 人喜欢椰菜和花椰菜，14 人喜欢这 3 种蔬菜。270 个学生中有多少人对这 3 种菜都不喜欢？
9. 一个学校有 507，292，312 和 344 个学生分别选了微积分、离散数学、数据结构和程序设计语言课，且有 14 人选了微积分和数据结构课，213 人选了微积分和程序设计语言课，211 人选了离散数学和数据结构课，43 人选了离散数学和程序设计语言课，没有学生同时选微积分和离散数学课，也没有学生同时选数据结构和程序设计语言课。问有多少学生在微积分、离散数学、数据结构或程序设计语言中选了课？
10. 求不超过 100 且不被 5 或 7 整除的正整数个数。

11. 求不超过 100 且是奇数或平方数的正整数个数。
12. 求不超过 1 000 且是平方数或立方数的正整数个数。
13. 有多少 8 位二进制串不包含 6 个连续的 0?
*14. 26 个英文字母的排列中有多少个不包含串 fish，rat 或 bird?
15. 在 10 个十进制数字的排列中以 3 个数字 987 开始，在第 5 和第 6 位包含数字 45，且最后 3 位是 123?
16. 有 4 个集合，每个集合有 100 个元素，每一对集合有 50 个公共元素，每 3 个集合有 25 个公共元素，并且有 5 个元素在所有的 4 个集合里。问在这 4 个集合的并集中有多少个元素?
17. 有 4 个集合，如果这些集合分别有 50，60，70 和 80 个元素，每一对集合有 5 个公共元素，每 3 个集合有 1 个公共元素，并且没有元素在所有的 4 个集合里。问在这 4 个集合的并集中有多少个元素?
18. 在容斥原理所给出的有关 10 个集合并集元素数的公式中有多少项?
19. 根据容斥原理写出关于 5 个集合并集元素数的显示公式。
20. 有 5 个集合，如果每个集合包含 10 000 个元素，每对集合包含 1 000 个公共元素，每 3 个集合包含 100 个公共元素，每 4 个集合包含 10 个公共元素，且这 5 个集合有 1 个公共元素。问在这些集合的并集中有多少个元素?
21. 有 6 个集合，如果知道其中任何 3 个集合都是不相交的，根据容斥原理写出关于这 6 个集合并集元素数的显示公式。
*22. 使用数学归纳法证明容斥原理。
23. 设 E_1，E_2 和 E_3 是样本空间 S 的 3 个事件。求一个关于 $E_1 \cup E_2 \cup E_3$ 的概率的公式。
24. 当一个硬币掷 5 次时头像向下恰好 3 次，第一次和最后一次头像向下，或第二次和第四次头像向上求其概率。
25. 从 1 到 100（含 1 和 100 在内）不允许重复地随机取 4 个数，求所有的都是奇数、所有的都被 3 整除或所有的都可被 5 整除的概率。
26. 一个样本空间有 4 个事件，如果其中没有 3 个事件同时出现，求关于这 4 个事件的并的概率公式。
27. 一个样本空间有 5 个事件，如果其中没有 4 个事件同时出现，求关于这 5 个事件的并的概率公式。
28. 一个样本空间有 n 个事件，如果其中没有 2 个事件同时出现，求关于这 n 个事件的并的概率公式。
29. 求一个样本空间中 n 个事件的并的概率公式。

5.6 容斥原理的应用

5.6.1 引言

可以使用容斥原理求解许多计数问题。例如，我们可以使用这个原理找出小于某个正整数的素数个数。通过计数从一个有穷集到另一个有穷集的映上函数的个数能够求解许多问题，而容斥原理就可以用来求出这种函数的个数。也可以使用容斥原理求解著名的帽子寄存

问题。帽子寄存问题是：一个招待随机地将帽子发还存放帽子的人，求没有人取回自己的帽子的概率。

5.6.2 容斥原理的另一种形式

容斥原理有另一种表述形式，它在计数问题中是很有用的。特别地，这种形式可以用于求解在一个集合中的元素数，使得这些元素不具有 n 个性质 P_1，P_2，…，P_n 中的任何一条性质。

设 A_i 是包含具有性质 P_i 的元素的子集。具有所有这些性质 P_{i_1}，P_{i_2}，…，P_{i_k} 的元素数将记作 $N(P_{i_1}P_{i_2}\cdots P_{i_k})$。用集合的术语写这些等式，有

$$|A_{i_1}\cap A_{i_2}\cap\cdots\cap A_{i_k}| = N(P_{i_1}P_{i_2}\cdots P_{i_k})$$

如果 n 个性质 P_1，P_2，…，P_n 中的任何一条都不具有的元素数记作 $N(P'_1P'_2\cdots P'_n)$，并且集合中的元素数记作 N，那么有

$$N(P'_1,P'_2\cdots P'_n) = N - |A_1\cup A_2\cup\cdots\cup A_n|$$

由容斥原理，有

$$N(P'_1P'_2\cdots P'_n) = N - \sum_{1\leqslant i\leqslant n} N(P_i) + \sum_{1\leqslant i<j\leqslant n} N(P_iP_j) - \sum_{1\leqslant i<j<k\leqslant n} N(P_iP_jP_k)$$
$$+\cdots+(-1)^n N(P_1P_2\cdots P_n)$$

下面的例子说明怎样使用容斥原理确定具有约束条件的方程的整数解的个数。

例 1 $x_1+x_2+x_3=11$ 有多少个整数解？其中 x_1，x_2 和 x_3 是非负整数，且 $x_1\leqslant 3$，$x_2\leqslant 4$，$x_3\leqslant 6$。

解 为使用容斥原理，令解的性质 P_1 为 $x_1>3$，性质 P_2 为 $x_2>4$，性质 P_3 为 $x_3>6$。满足不等式 $x_1\leqslant 3$，$x_2\leqslant 4$ 以及 $x_3\leqslant 6$ 的解的个数是

$$N(P'_1P'_2P'_3) = N - N(P_1) - N(P_2) - N(P_3) + N(P_1P_2)$$
$$+ N(P_1P_3) + N(P_2P_3) - N(P_1P_2P_3)$$

使用与 4.6 节例 6 相同的技术，得

N = 解的总数 $= C(3+11-1,11) = 78$

$N(P_1)$ = (具有 $x_1\geqslant 4$ 的解数) $= C(3+7-1,7) = C(9,7) = 36$

$N(P_2)$ = (具有 $x_2\geqslant 5$ 的解数) $= C(3+6-1,6) = C(8,6) = 28$

$N(P_3)$ = (具有 $x_3\geqslant 7$ 的解数) $= C(3+4-1,4) = C(6,4) = 15$

$N(P_1P_2)$ = (具有 $x_1\geqslant 4$ 且 $x_2\geqslant 5$ 的解数) $= C(3+2-1,2) = C(4,2) = 6$

$N(P_1P_3)$ = (具有 $x_1\geqslant 4$ 且 $x_3\geqslant 7$ 的解数) $= C(3+0-1,0) = 1$

$N(P_2P_3)$ = (具有 $x_2\geqslant 5$ 且 $x_3\geqslant 7$ 的解数) $= 0$

$N(P_1P_2P_3)$ = (具有 $x_1\geqslant 4, x_2\geqslant 5$ 且 $x_3\geqslant 7$ 的解数) $= 0$

把这些等式代入关于 $N(P'_1P'_2P'_3)$ 的公式，证明满足 $x_1\leqslant 3$，$x_2\leqslant 4$ 以及 $x_3\leqslant 6$ 的解的个数等于

$$N(P'_1P'_2P'_3) = 78-36-28-15+6+1+0-0 = 6$$ ■

5.6.3 伊拉脱森筛

可以用容斥原理找出不超过一个给定正整数的素数个数。一个合数可以被一个不超过它的平方根的素数整除。因此，为找出不超过 100 的素数个数，首先注意到不超过 100 的合数一定有一个不超过 10 的素因子。由于小于 10 的素数只有 2，3，5 和 7，因此不超过 100 的素数就是这 4 个素数以及那些大于 1 和不超过 100 且不被 2，3，5 或 7 整除的正整数。为应用容斥原理，令 P_1 是一个整数被 2 整除的性质，P_2 是一个整数被 3 整除的性质，P_3 是一个整数被 5 整除的性质，P_4 是一个整数被 7 整除的性质。于是，不超过 100 的素数个数是

$$4+N(P'_1P'_2P'_3P'_4)$$

由于存在 99 个比 1 大且不超过 100 的正整数，容斥原理显示

$$\begin{aligned}N(P'_1P'_2P'_3P'_4)=99&-N(P_1)-N(P_2)-N(P_3)-N(P_4)\\&+N(P_1P_2)+N(P_1P_3)+N(P_1P_4)+N(P_2P_3)\\&+N(P_2P_4)+N(P_3P_4)-N(P_1P_2P_3)-N(P_1P_2P_4)\\&-N(P_1P_3P_4)-N(P_2P_3P_4)+N(P_1P_2P_3P_4)\end{aligned}$$

不超过100(且大于 1)与被{2,3,5,7}的子集中的所有素数整除的正整数个数是$\lfloor 100/N\rfloor$,其中 N 是这个子集中的素数之积（这是由于任意两个素数都没有公因子)。因此

$$\begin{aligned}N(P'_1P'_2P'_3P'_4)=99&-\left\lfloor\frac{100}{2}\right\rfloor-\left\lfloor\frac{100}{3}\right\rfloor-\left\lfloor\frac{100}{5}\right\rfloor-\left\lfloor\frac{100}{7}\right\rfloor\\&+\left\lfloor\frac{100}{2\cdot 3}\right\rfloor+\left\lfloor\frac{100}{2\cdot 5}\right\rfloor+\left\lfloor\frac{100}{2\cdot 7}\right\rfloor+\left\lfloor\frac{100}{3\cdot 5}\right\rfloor+\left\lfloor\frac{100}{3\cdot 7}\right\rfloor+\left\lfloor\frac{100}{5\cdot 7}\right\rfloor\\&-\left\lfloor\frac{100}{2\cdot 3\cdot 5}\right\rfloor-\left\lfloor\frac{100}{2\cdot 3\cdot 7}\right\rfloor-\left\lfloor\frac{100}{2\cdot 5\cdot 7}\right\rfloor-\left\lfloor\frac{100}{3\cdot 5\cdot 7}\right\rfloor\\&+\left\lfloor\frac{100}{2\cdot 3\cdot 5\cdot 7}\right\rfloor\\&=99-50-33-20-14+16+10+7+6+4+2-3-2-1-0+0\\&=21\end{aligned}$$

于是存在 4＋21＝25 个不超过 100 的素数。

可以用伊拉脱森[1]筛求不超过一个给定正整数的所有的素数。例如，可用下面的过程找不超过 100 的所有素数。首先，保留 2 而将其余那些被 2 整除的整数删除。因为 3 是保留下来的第一个大于 2 的整数，除 3 之外，删除其余那些被 3 整除的整数。因为 5 是在 3 后面下一个留下来的整数，除 5 之外删除其余那些被 5 整除的整数。下一个留下的整数是 7，因此留下 7，删除其余那些被 7 整除的整数。由于所有不超过 100 的合数被 2，

[1] 伊拉脱森（Eratosthens，公元前 276—194）　只知道伊拉脱森诞生在塞伦，埃及西部的一个希腊人聚居地，并且在雅典的柏拉图研究院从事研究。我们也知道托勒密二世皇帝曾邀请伊拉脱森到亚力山大给他的儿子做家庭教师并且后来伊拉脱森成为著名的亚力山大图书馆馆长，这个图书馆是古代智慧的宝库。伊拉脱森是一个多才多艺的学者，他的著作涉及数学、地理学、天文学、历史、哲学和文学评论诸方面。此外他还研究数学，他最引人注目的是关于古代历史编年表和地球大小测量的工作。

3，5 或 7 整除，那么所有留下来的大于 1 的数是素数。在表 5-2 中，4 个表显示了每一步删除的整数，在第一个表中得到下划线的是除 2 之外其余的被 2 整除的整数，在第二个表中得到下划线的是除 3 之外其余的被 3 整除的整数，第三个表得到下划线的是除 5 之外其余的被 5 整除的整数，在第四个表中得到下划线的是除 7 之外其余的被 7 整除的整数。没有下划线的整数是不超过 100 的素数。

表 5-2 伊拉脱森筛

除 2 之外其余被 2 整除的整数得到下划线

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

除 3 之外其余被 3 整除的整数得到下划线

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

除 5 之外其余被 5 整除的整数得到下划线

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

除 7 之外被 7 整除的整数得到下划线

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

5.6.4 映上函数的个数

也可以用容斥原理确定从 m 元素集合到 n 元素集合的映上函数的个数。首先考虑下面的例子。

例 2 从 6 元素集合到 3 元素集合有多少个映上函数？

解 假定在陪域中的元素是 b_1，b_2，b_3。设 P_1，P_2，P_3 分别是 b_1、b_2，b_3 不在函数值域中的性质。注意到一个函数是映上的，当且仅当没有性质 P_1、P_2 和 P_3。根据容斥原理得到 6 元素集合到 3 元素集合的映上函数个数是

$$N(P'_1P'_2P'_3) = N - [N(P_1) + N(P_2) + N(P_3)] + [N(P_1P_2) + N(P_1P_3) + N(P_2P_3)] - N(P_1P_2P_3)$$

其中 N 是从 6 元素集合到 3 元素集合的函数总数。我们将对等式右边的每一项求值。

由 4.1 节的例 8 得出 $N = 3^6$。注意到 $N(P_i)$ 是值域中不含 b_i 的函数个数。所以，对于

定义域中的每个元素的函数值有 2 种选择。从而得到$N(P_i)=2^6$。此外，这种项有 $C(3,1)$ 个。注意到$N(P_iP_j)$是值域中不含 b_i 和b_j 的函数个数。所以，对于定义域中的每个元素的函数值只有 1 种选择。从而得到$N(P_iP_j)=1^6=1$ 。此外，这种项有$C(3,2)$个。还有，注意到$N(P_1P_2P_3)=0$ ，因为这个项是值域中不含 b_1，b_2 和 b_3 的函数个数。很清楚，没有这样的函数。于是，从 6 元素集合到 3 元素集合的映上函数个数是

$$3^6-C(3,1)2^6+C(3,2)1^6=729-192+3=540$$ ■

现在叙述从 m 元素集合到n 元素集合的映上函数个数的一般性结果。这个结果的证明留给读者作为练习。

定理 1 设 m 和 n 是正整数，$m\geq n$。那么存在

$$n^m-C(n,1)(n-1)^m+C(n,2)(n-2)^m-\cdots+(-1)^{n-1}C(n,n-1)1^m$$

个从 m 元素集合到n 元素集合的映上函数。

下面给出定理 1 的另一个应用的实例。

例 3 把 5 项工作分给 4 个不同的雇员，如果每个雇员至少分配 1 项工作，问有多少种方式?

解 把工作分配看作从 5 个工作集合到 4 个雇员集合的函数。每个雇员至少得到 1 项工作的分配对应于从工作集合到雇员集合的映上函数。因此，由定理 1 存在

$$4^5-C(4,1)3^5+C(4,2)2^5-C(4,3)1^5=1\,024-972+192-4=240$$

种方式来分配工作，使得每个雇员至少得到 1 项工作。 ■

5.6.5 错位排列

下面将用容斥原理计数排列 n 个物体并使得没有一个物体在它的初始位置上的方式数。考虑下面的例子。

例 4 帽子寄存问题。在一个餐厅里一个新的雇员寄存 n 个人的帽子时忘记把寄存号放在帽子上。当顾客取回他们的帽子时，这个雇员从剩下的帽子中随机选择发给他们。问没有一个人收到自己的帽子的概率是多少? ■

注意 答案就是重新排列帽子使得没有帽子在它的初始位置上的方式数除以 n 个帽子的排列数$n!$。在我们找出排列 n 个物体并使得没有一个物体在它的初始位置上的方式数以后再考虑这个例子。

一个错位排列[㊀]是使得没有一个物体在它的初始位置上的排列。为求解例 4 中的问题我们需要确定 n 个物体的错位排列数。

㊀ 历史注记：在一个古老的法国纸牌相遇(匹配)游戏中，一套 52 张牌摆成一行。摆放第二套牌使得其中每张牌放在第一套牌的某一张的顶部。通过统计在两套牌中匹配的牌数来确定得分。在 1708 年，皮埃尔·雷蒙德·蒙特莫特(1678—1719)提出了"相遇问题"：在相遇游戏中没有匹配发生的概率是多少？蒙特莫特问题的解是随机选择 52 个物体的排列恰为错位排列的概率，即 $D_{52}/52!$，正如我们将看到的这个概率近似为 $1/e$。

例 5 排列 21 453 是 12 345 的一个错位排列，因为没有数在它的初始位置上。但是，21 543 不是 12 345 的错位排列，因为 4 留在它的初始位置上。 ■

令 D_n 表示 n 个物体的错位排列数。例如，$D_3=2$，因为 123 的错位排列是 231 和 312。我们将使用容斥原理对所有的正整数 n 求 D_n。

定理 2 n 元素集合的错位排列数是

$$D_n = n!\left[1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right]$$

证 如果排列保持元素 i 不变，就设排列有性质 P_i。错位排列的个数就是对 $i=1,2,\cdots,n$，没有性质 P_i 的排列数，或

$$D_n = N(P'_1P'_2\cdots P'_n)$$

使用容斥原理得到

$$D_n = N-\sum_i N(P_i)+\sum_{i<j}N(P_iP_j)-\sum_{i<j<k}N(P_iP_jP_k)+\cdots+(-1)^nN(P_1P_2\cdots P_n)$$

其中 N 是 n 个元素的排列数。这个等式说明，所有的元素都发生变化的排列数，等于排列的总数减去至少保持 1 个元素不变的排列数，加上至少保持 2 元素不变的排列数，减去至少保持 3 个元素不变的排列数，等等。现在找出在等式右边出现的所有的量。

首先注意到 $N=n!$，因为 N 仅仅就是 n 个元素排列的总数。还有，$N(P_i)=(n-1)!$。这是由乘积法则得到的，因为 $N(P_i)$ 是保持元素 i 不变的排列数，因而第 i 个位置是确定的，但是其余的每个位置可以放任意元素。类似地，

$$N(P_iP_j)=(n-2)!$$

因为这是保持元素 i 和 j 不变的排列数，但是其余 $(n-2)$ 元素的位置可以被任意地安排。一般说来有

$$N(P_{i_1}P_{i_2}\cdots P_{i_m})=(n-m)!$$

因为这是保持元素 i_1，i_2，$\cdots$，i_m 不变的排列数，但是其他 $(n-m)$ 个元素的位置可以被任意安排。由于存在 $C(n,m)$ 种方式从 n 个元素中选择 m 个，从而有

$$\sum_{1\le i\le n}N(P_i) = C(n,1)(n-1)!$$

$$\sum_{1\le i<j\le n}N(P_iP_j) = C(n,2)(n-2)!$$

一般地有

$$\sum_{1\le i_1<i_2<\cdots<i_m\le n}N(P_{i_1}P_{i_2}\cdots P_{i_m}) = C(n,m)(n-m)!$$

所以，把这些等式代入关于 D_n 的公式，得

$$\begin{aligned}D_n &= n!-C(n,1)(n-1)!+C(n,2)(n-2)!-\cdots+(-1)^nC(n,n)(n-n)!\\ &= n!-\frac{n!}{1!\,(n-1)!}(n-1)!+\frac{n!}{2!\,(n-2)!}(n-2)!-\cdots+(-1)^n\frac{n!}{n!\,0!}0!\end{aligned}$$

化简这个表达式，得

$$D_n = n!\left[1-\frac{1}{1!}+\frac{1}{2!}-\cdots+(-1)^n\frac{1}{n!}\right]$$

□

现在对于给定的正整数 n 求 D_n 就简单了。例如，使用定理 2，得

$$D_3 = 3!\left[1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}\right]=6\left(1-1+\frac{1}{2}-\frac{1}{6}\right)=2$$

正如我们前面所看到的。

现在可以给出例 4 中的问题的解。

解 没有一个人收到自己的帽子的概率是 $D_n/n!$。由定理 2，这个概率是

$$\frac{D_n}{n!}=1-\frac{1}{1!}+\frac{1}{2!}-\cdots+(-1)^n\frac{1}{n!}$$

对于 $2\leqslant n\leqslant 7$，这个概率的值在表 5-3 中给出。

表 5-3 错位排列的概率

n	2	3	4	5	6	7
$D_n/n!$	0.50000	0.33333	0.37500	0.36667	0.36806	0.36786

使用微积分的方法可以证明

$$e^{-1}=1-\frac{1}{1!}+\frac{1}{2!}-\cdots+(-1)^n\frac{1}{n!}+\cdots\approx 0.368$$

因为这是一个项趋向于 0 的交错级数，当 n 无限增长时，没有一个人取回自己的帽子的概率趋于 $e^{-1}\approx 0.368$。事实上，可以证明这个概率与 e^{-1} 的差在 $1/(n+1)!$ 之内。 ■

练习

1. 假设 1 蒲式耳 100 个苹果中 20 个有虫，15 个有擦伤。只有无虫也没擦伤的苹果才可以卖。如果 10 个擦伤的苹果有虫，那么 100 个苹果中有多少个可以卖？
2. 1 000 个人申请喜马拉雅山登山旅游，450 人有高山病，622 人不是处在很好的状态，30 人有过敏症。一个申请人当且仅当没有高山病，并且处在一个良好的状态下和没有过敏症，才算合格。如果 111 个申请人有高山病且不是在良好状态，14 人有高山病和过敏症，18 人不是在一个良好的状态并且有过敏症。9 个人有高山病并且不是在良好状态和有过敏症，那么有多少申请人合格？
3. 方程 $x_1+x_2+x_3=13$ 有多少个解？其中 x_1，x_2，x_3 是小于 6 的非负整数。
4. 求方程 $x_1+x_2+x_3+x_4=17$ 的解的个数，其中 x_i $(i=1,2,3,4)$ 是非负整数，满足条件 $x_1\leqslant 3$，$x_2\leqslant 4$，$x_3\leqslant 5$，$x_4\leqslant 8$。
5. 使用容斥原理求小于 200 的素数个数。
6. 一个整数叫作*无平方的*如果它不被一个大于 1 的正整数的平方整除。求小于 100 的无平方的正整数个数。
7. 有多少个小于 10 000 的正整数不是一个整数的 2 次或更高次幂？
8. 从 7 元素集合到 5 元素集合有多少个映上函数？

9. 有多少种方式把 6 个不同的玩具分给 3 个不同的孩子，使得每个孩子至少得到 1 个玩具？
10. 把 8 个不同的球放入 3 个不同的罐子，如果每个罐子至少有 1 个球，有多少种方法？
11. 有多少种方式把 7 项不同的工作分给 4 个不同的雇员，使得每个雇员至少得到 1 项工作，并且把最困难的工作分给最好的雇员？
12. 列出 {1, 2, 3, 4} 的所有的错位排列。
13. 一个 7 元素集合有多少个错位排列？
14. 如果寄存帽子的人随机发回帽子，10 个人中没有一个人得到他自己帽子的概率是多少？
15. 一个把信放人信袋的机器发生了故障并且随机把信放人信袋中。在一组 100 封信中发生下面事件的概率是多少？
   a) 没有信放对了信袋。
   b) 恰好 1 封信放对了信袋。
   c) 恰好 98 封信放对了信袋。
   d) 恰好 99 封信放对了信袋。
   e) 所有的信都放对了信袋。
16. 在同一个教室为一组 n 个学生分配两个班的座位。如果没有学生在两个班分到同一个座位，有多少种方式？
*17. 有多少种方式安排数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，使得没有偶数在它的初始位置上？
*18. 设 D_n 表示 n 个物体的错位排列数，对于 $n\geqslant 2$，用组合论证证明序列 $\{D_n\}$ 满足递推关系

$$D_n=(n-1)(D_{n-1}+D_{n-2})$$

*19. 使用练习 18 证明，对于 $n\geqslant 1$，

$$D_n=nD_{n-1}+(-1)^n$$

20. 使用练习 19 求关于 D_n 的显式公式。
21. 对哪些正整数 n，错位排列数 D_n 是偶数？
22. 假设 p 和 q 是不同的素数。使用容斥原理求 ϕ (pq)，即不超过 pq 且与 pq 互素的整数个数。
*23. 当 n 的素因子分解式是

$$n=p_1^{a_1}p_2^{a_2}\cdots p_m^{a_m}$$

使用容斥原理推导一个关于$\phi(n)$的公式。
*24. 证明如果 n 是正整数，那么

$$n!=C(n,0)D_n+C(n,1)D_{n-1}+\cdots+C(n,n-1)D_1+C(n,n)D_0$$

其中 D_k 是 k 个物体的错位排列数。
25. 有多少个 {1, 2, 3, 4, 5, 6} 的错位排列的前 3 位是取某种次序的整数 1, 2, 3？
26. 有多少个 {1, 2, 3, 4, 5, 6} 的错位排列的后 3 位是取某种次序的整数 1, 2, 3？
27. 证明定理 1。

关键术语和结果

术语

递推关系：一个公式，它把序列除了某些初始项以外的项，表示成这个序列前面的一个或若

于个项的函数

递推关系的初始条件：满足递推关系的序列在该关系起作用之前的某些项的值

常系数线性齐次递推关系：一个递推关系，除了初始项之外，它把序列的项表示成前面项的线性组合

常系数线性齐次递推关系的特征根：与常系数线性齐次递推关系相关的多项式的根

常系数线性非齐次递推关系：一个递推关系，除了初始项之外，它把序列的项表示成前面项的线性组合加上一个仅仅依赖于序标的不恒为 0 的函数

分而治之算法：求解问题的一种算法，求解中把问题递归地划分成固定数目的较小的同种类型的问题

序列的生成函数：用序列的第 n 项作为 x^n 的系数的形式幂级数

伊拉脱森筛：找出小于一个给定正整数的素数的过程

错位排列：使得没有物体处在它的初始位置上的排列

结果

两个有穷集合并集的元素个数公式：

$$|A\cup B| = |A| + |B| - |A\cap B|$$

三个有穷集合并集的元素个数公式：

$$|A\cup B\cup C| = |A| + |B| + |C| - |A\cap B| - |A\cap C| - |B\cap C| + |A\cap B\cap C|$$

容斥原理：

$$|A_1\cup A_2\cup\cdots\cup A_n| = \sum_{1\leqslant i\leqslant n}|A_i| - \sum_{1\leqslant i<j\leqslant n}|A_i\cap A_j| + \sum_{1\leqslant i<j<k\leqslant n}|A_i\cap A_j\cap A_k| - \cdots + (-1)^{n+1}|A_1\cap A_2\cap\cdots\cap A_n|$$

从 m 元素集合到 n 元素集合的映上函数个数：

$$n^m - C(n,1)(n-1)^m + C(n,2)(n-2)^m - \cdots + (-1)^{n-1}C(n,n-1)1^m$$

n 个物体的错位排列数：

$$D_n = n!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n\frac{1}{n!}\right]$$

复习题

1. a）什么是递推关系？
   b）如果在一个获利 9% 的账上储蓄 1 000 000 美元，求与 n 年后账上钱数有关的递推关系。
2. 解释怎样用斐波那契数求解关于兔子的斐波那契问题。
3. a）找出与求解汉诺塔难题所需的步数有关的递推关系。
   b）显示怎样使用迭代来求解这个递推关系。
4. a）解释怎样找一个与不包含两个连续的 1 的 n 位二进制串个数有关的递推关系。
   b）描述另一个计数问题使得它的解满足同一个递推关系。
5. 定义一个 k 阶的线性齐次递推关系。

6. a) 解释怎样求解二阶线性齐次递推关系。
   b) 如果 $a_0=3$，$a_1=15$，对于 $n\geqslant2$，求解递推关系 $a_n=13a_{n-1}-22a_{n-2}$。
   c) 如果 $a_0=3$，$a_1=35$，对于 $n\geqslant2$，求解递推关系 $a_n=14a_{n-1}-49a_{n-2}$。
7. a) 如果 $f(n)$ 满足分而治之递推关系 $f(n)=af(n/b)+g(n)$，这里 b 整除正整数 n，解释怎样求 $f(b^k)$，其中 k 是正整数。
   b) 如果 $f(n)=3f(n/4)+5n/4$ 且 $f(1)=7$，求 $f(256)$。
8. a) 对于用二分检索在表中找一个数所用的比较次数，推导一个分而治之的递推关系。
   b) 从你在 a) 给出的分而治之的递推关系，使用 5.3 节中的定理 1，对于二分检索所用的比较次数给出一个大 O 估计。
9. a) 给出一个关于 3 个集合并集元素个数的公式。
   b) 解释为什么这个公式是有效的。
   c) 解释怎样使用 a) 中的公式求不超过 1000 且能被 6，10 或 15 整除的正整数的个数。
   d) 解释怎样使用 a) 的公式求方程 $x_1+x_2+x_3+x_4=22$ 的非负整数解的个数，其中 $x_1<8$，$x_2<6$，$x_3<5$。
10. a) 给出一个关于 4 个集合并集元素个数的公式。解释为什么它是有效的。
    b) 假设 A_1，A_2，A_3 和 A_4 每个集合含 25 个元素，其中任何 2 个集合的交含 5 个元素，任何 3 个集合的交含 2 个元素，所有 4 个集合含 1 个公共元素。问在这 4 个集合的并集中有多少个元素？
11. a) 叙述容斥原理。
    b) 概述这个原理的证明。
12. 解释怎样使用容斥原理计数从 m 元素集合到 n 元素集合的映上函数的个数。
13. a) 怎样计数把 m 项工作分给 n 个雇员并使得每个雇员至少得到一项工作的方案数？
    b) 把 7 项工作分给 3 个雇员并使得每个雇员至少得到一项工作有多少种方案？
14. 解释怎样使用容斥原理计数不超过正整数 n 的素数个数。
15. a) 定义一个错位排列。
    b) 一个寄存帽子的人给 n 个人发还帽子并使得没有人得到自己帽子的方式的计数，为什么和 n 个物体的错位排列数一样？
    c) 解释怎样计数 n 个物体的错位排列数。

补充练习

1. 一个 10 人小组开始一系列的通信活动，每个人把这封信寄给另外 4 个人。每个收到信的人再把这封信寄给另外的 4 个人。
   a) 如果没有人收到 2 封以上的信，求与这个通信活动的第 n 步寄出信数有关的递推关系。
   b) 在 a) 中递推关系的初始条件是什么？
   c) 在这个通信活动的第 n 步寄出了多少封信？
2. 一个核反应堆产生了 18 克放射性同位素。每小时放射性同位素衰变 1%。
   a) 对 n 小时后留下的同位素量建立一个递推关系。
   b) 对于 a) 中的递推关系，初始条件是什么？

c）求解这个递推关系。

3. 美国政府每小时印 1 美元纸币超过 10 000 张，5 美元纸币超过 4 000 张，10 美元纸币超过 3 000 张，20 美元纸币超过 2 500 张，50 美元纸币超过 1 000 张，100 美元纸币与前一小时的张数一样。在初始时刻每种钱币有 1 000 张。

   a）建立一个关于第 n 小时总钱数的递推关系。

   b）对于 a）中的递推关系，初始条件是什么？

   c）求解这个关于第 n 小时总钱数的递推关系。

   d）建立一个关于前 n 小时总钱数的递推关系。

   e）求解这个关于前 n 小时总钱数的递推关系。

4. 每个前一小时已经存在的细菌在每小时都分裂出两个新的细菌，并且所有的细菌只有 2 小时的寿命。假设这群细菌开始时有 100 个新细菌。

   a）建立关于 n 小时后存在细菌数目的递推关系。

   b）这个递推关系的解是什么？

   c）什么时候这群细菌的个数将超过 100 万个？

5. 使用两个不同的信号在通信信道发送信息。传递一个信号需要 2 微秒，传送另一个信号需要 3 微秒。一个信息的每个信号紧跟着下一个信号。

   a）求与在 n 微秒中可以发送的不同信号数有关的递推关系。

   b）对于 a）中的递推关系，初始条件是什么？

   c）在 12 微秒内可以发送多少个不同的信息？

6. 一个小邮局只有 4 分、6 分和 10 分邮票。如果考虑邮票使用的次序，求与这些邮票构成 n 分邮费的方式数有关的递推关系。这个递推关系的初始条件是什么？

7. 使用在练习 6 描述的规则有多少种方式构成下述邮资？

   a）12 分　　b）14 分

   c）18 分　　d）22 分

8. 求具有 $a_0=1$ 和 $b_0=2$ 的联立方程组。

   a）$a_n=a_{n-1}+b_{n-1}$

   b）$b_n=a_{n-1}-b_{n-1}$

9. 如果 $a_0=1$ 和 $a_1=2$，求解递推关系 $a_n=a_{n-1}^2/a_{n-2}$。［提示：两边取对数得到关于序列 $\log a_n$ 的递推关系，$n=0, 1, 2, \cdots$。］

*10. 如果 $a_0=2$ 和 $a_1=2$，求解递推关系 $a_n=a_{n-1}^3a_{n-2}^2$。（见练习 9 的提示。）

11. 如果 $a_0=2$，$a_1=4$ 和 $a_2=8$，求解递推关系 $a_n=3a_{n-1}-3a_{n-2}+a_{n-3}+1$。

12. 如果 $a_0=2$，$a_1=2$ 和 $a_2=4$，求解递推关系 $a_n=3a_{n-1}-3a_{n-2}+a_{n-3}$。

*13. 假设在 5.1 节的例 4 中一对兔子在繁殖 2 次以后就离开这个岛。求与第 n 个月中的岛上兔子对数有关的递推关系。

14. 当 n 可被 5 整除时求解递推关系 $f(n)=3f(n/5)+2n^4$，其中 $n=5^k$，k 是正整数，$f(1)=1$。

15. 如果 f 是增函数，估计练习 14 中 f 的大小。

16. 找出与下述算法所使用比较次数有关的递推关系：通过把 n 个数的序列递归地划分成两个子序列找出最大和第二大的元素，在每一步划分时要求这两个子序列项数相等或一个

子序列比另一个子序列多一项。当子序列达到 2 项时停止。

17. 估计练习 16 描述的算法所使用的比较次数。

设$\{a_n\}$是实数序列。这个序列的前项差分定义如下：第一个前项差分是 $\Delta a_n = a_{n+1} - a_n$；第 $k+1$ 个前项差分 $\Delta^{k+1}a_n$ 是通过 $\Delta^{k+1}a_n = \Delta^k a_{n+1} - \Delta^k a_n$ 由 $\Delta^k a_n$ 得到的。

18. 求 $\Delta^k a_n$，其中
   a) $a_n = 3$
   b) $a_n = 4n + 7$
   c) $a_n = n^2 + n + 1$

19. 设 $a_n = 3n^3 + n + 2$。求 $\Delta^k a_n$，其中 k 等于
   a) 2　　b) 3　　c) 4

*20. 假设$a_n = P(n)$，其中 P 是d 次多项式。证明对所有的非负整数 n，$\Delta^{d+1}a_n = 0$。

21. 令$\{a_n\}$和$\{b_n\}$是实数序列。证明
   $\Delta(a_n b_n) = a_{n+1}(\Delta b_n) + b_n(\Delta a_n)$

22. 证明如果$F(x)$和 $G(x)$分别是序列$\{a_k\}$和$\{b_k\}$的生成函数，且 c 和d 是实数，那么$\{cF(x) + dG(x)\}$是$\{ca_k + db_k\}$的生成函数。

23. (需要微积分) 这个练习说明了怎样使用生成函数求解递推关系$(n+1)a_{n+1} = a_n + (1/n!)$，$n \geqslant 0$，初始条件 $a_0 = 1$。
   a) 设$G(x)$是关于$\{a_n\}$的生成函数。证明$G'(x) = G(x) + e^x$ 且 $G(0) = 1$。
   b) 由 a) 证明$(e^{-x}G(x))' = 1$，且断定 $G(x) = xe^x + e^x$。
   c) 使用 b) 找出关于 a_n 的封闭公式。

24. 假设在离散数学班的第一次考试中 14 个学生得 A，第二次考试中 18 个得 A。如果 22 个学生在第一或第二次考试得 A，有多少学生两次考试都得 A？

25. 在蒙默思郡（英国威尔士郡原郡名）323 个农场至少有马、牛、羊中的 1 种。如果 224 个农场有马，85 个有牛，57 个有羊，18 个农场 3 种家畜全有，那么有多少个农场恰好有这 3 种家畜中的 2 种？

26. 查询某学院关于学生记录的数据库得到下述数据：学院有 2 175 个学生，其中 1 675 个不是一年级学生，1 074 个学生选了微积分，444 个学生选了离散数学，607 个不是一年级学生且选了微积分，350 个学生选了微积分和离散数学，201 个不是一年级学生且选了离散数学，143 个不是一年级学生并且选了微积分和离散数学。所有这些对查询的回答都是正确的吗？

27. 某大学数学学院的学生可以选择下述一个或多个方向作为主修方向：应用数学(AM)，纯粹数学(PM)，运筹学(OR)及计算机科学(CS)。如果包括同时主修在内，主修 AM 的有 23 个学生，主修 PM 的有 17 个学生，主修 OR 的 44 个，主修 CS 的 63 个，主修 AM 与 PM 的 5 个，主修 AM 与 CS 的 8 个，主修 AM 与 OR 的 4 个，主修 PM 与 CS 的 6 个，主修 PM 与 OR 的 5 个；主修 OR 与 CS 的 14 个；主修 PM，OR 与 CS 的 2 个，主修 AM，OR 与 CS 的 2 个，主修 PM，AM 与 OR 的 1 个，主修 PM，AM 与 CS 的 1 个，还有 1 个主修所有 4 个方向。问这个学院有多少学生？

28. 当使用容斥原理表示 7 个集合的并集中的元素个数时，如果其中没有 6 个或更多的集合

含有公共元素，那么需要多少项？

29. 方程 $x_1+x_2+x_3=20$ （$2<x_1<6$，$6<x_2<10$，$0<x_3<5$）有多少个正整数解？

30. 有多少个小于 1 000 000 的正整数
    a）能被 2，3 或 5 整除？
    b）不被 7，11 或 13 整除？
    c）能被 3 但不被 7 整除？

31. 有多少个小于 200 的正整数是
    a）整数的 2 次或更高次幂？
    b）整数的 2 次或更高次幂，或是素数？
    c）不被一个大于 1 的整数的平方整除？
    d）不被一个大于 1 的整数的立方整除？
    e）不被 3 个或更多的素数整除？

*32. 把 6 个不同的工作分给 3 个不同的雇员，如果最难的工作分给最有经验的雇员并且最容易的工作分给最缺少经验的雇员，那么有多少种分法？

33. 由寄存帽子的人随机发还给 n 个人的帽子，那么恰好一个人拿到自己帽子的概率是多少？

34. 有多少个 6 位二进制串不包含 4 个连续的 1？

35. 一个 6 位二进制串包含至少 4 个 1 的概率是多少？

计算机题目

用下面的输入和输出写程序。

1. 给定正整数 n，列出汉诺塔难题从一根柱子到另一根柱子依照游戏规则移动 n 个盘子需要的所有移动。
2. 给定正整数 n 和整数 k，$1\leqslant k\leqslant n$，列出富雷姆－斯图尔特算法（见 5.1 节练习 48 前面的说明）依照游戏规则用 4 根柱子从一根柱子到另一根柱子移动 n 个盘子需要的所有移动。
3. 给定正整数 n，列出不包含连续 2 个 0 的所有的 n 位二进制序列。
4. 给定正整数 n，写出在 $n+1$ 个变量的乘积中加括号的所有方式。
5. 给定递推关系 $a_n=c_1a_{n-1}+c_2a_{n-2}$和正整数 k，其中 c_1 和 c_2 是实数，初始条件为 $a_0=C_0$ 和 $a_1=C_1$，使用迭代求 a_k。
6. 给定递推关系 $a_n=c_1a_{n-1}+c_2a_{n-2}$和初始条件 $a_0=C_0$ 和 $a_1=C_1$，确定唯一的解。
7. 给定形如$f(n)=af(n/b)+c$ 的递推关系，其中 a 是实数，b 是正整数，c 是实数，并给出正整数 k，使用迭代求$f(b^k)$。
8. 给定 3 个集合的交集中的元素个数，每两个集合的交集中的元素个数，和每个集合中的元素个数，求其并集中的元素个数。
9. 给定正整数 n，求在 n 个集合的并集中元素个数的公式。
10. 给定正整数 m 和 n，求从 m 元素集合到 n 元素集合的映上函数个数。
11. 给定正整数 n，列出集合 $\{1,2,3,\cdots,n\}$ 的所有错位排列。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 求 f_{100}，f_{500}和 f_{1000}的精确值，其中 f_n 是斐波那契数。
2. 求比 1 000 000 大、比 1 000 000 000 大和比 1 000 000 000 000 大的最小的斐波那契数。
3. 求尽可能多的同为素数的斐波那契数，目前还不知道是否存在无限多个这样的数。
4. 写出求解 10 个盘子的汉诺塔难题所需要的所有的移动。
5. 依照雷夫难题的规则，用 4 根柱子从一根柱子到另一根柱子移动 20 个盘子，写出用富雷姆－斯图尔特算法需要的所有移动。
6. 通过下面的方法验证求解 n 个盘子的雷夫难题的富雷姆猜想：对于尽可能多的整数 n，证明这个难题不可能使用比具有最优选择 k 的富雷姆-斯图尔特算法还要少的移动来求解。
7. 计算对于各种整数 n，包括 16，64，256 和 1024，使用在 5.3 节描述的快速乘法和整数相乘的标准算法（2.4 节算法 4）做两个 n 位整数相乘所需要的运算次数。
8. 计算对于各种整数 n，包括 4，16，64 和 128，使用在 5.3 节描述的快速矩阵乘法和矩阵相乘的标准算法（2.6 节算法 1）做两个 $n\times n$ 矩阵相乘所需要的运算次数。
9. 使用伊拉脱森筛求不超过 1 000 的所有素数。
10. 使用在 5.6 节描述的求不超过 100 的素数个数的方法求不超过 10 000 的素数个数。
11. 列出 {1，2，3，4，5，6，7，8} 的所有的错位排列。
12. 对所有不超过 20 的正整数 n，计算 n 个物体的一个排列是错位排列的概率，并确定这些概率逼近 e^{-1}的速度。

写作题目

用课本以外的资料按下列要求写成短文。

1. 找出斐波那契提出的关于兔子数模型难题的原始材料。讨论斐波那契提出的这个问题和其他问题，并且给出关于斐波那契本人的某些信息。
2. 解释斐波那契数怎样在其他应用中出现，如叶序，植物叶片排列的研究，镜子反射的研究等等。
3. 描述汉诺塔难题的各种不同的变形问题，包括多于 3 个柱子的（包括课本和练习中讨论的雷夫难题在内），盘子移动受限制的，以及允许有同样大小盘子的。关于求解每种变形问题所要求的移动次数有什么已知的结论？
4. 尽可能多地讨论出现卡特朗数的不同的问题。
5. 查询幸运数（lucky number）的定义。解释怎样使用类似于伊拉脱森筛的技术找出幸运数。求所有小于 1 000 的幸运数。
6. 描述在数论中使用的筛法。使用这种方法已经得到了哪些结果？
7. 查询古代法国纸牌相遇游戏的规则。描述这些规则并且描述皮埃尔·雷蒙德·蒙特莫特关于“相遇问题”的论文。
8. 描述怎样使用指数生成函数求解各种计数问题。

9. 描述计数的 Polyá 理论和可使用这个理论求解的计数问题的种类。

10. 管家问题是求安排 n 对夫妇围圆桌就座的方法数，使得就座时男女相间并且没有丈夫和妻子相邻。解释怎样用卢卡斯(E. Lucas)方法求解这个问题。

11. 解释怎样使用棋盘多项式(rook polynomial)求解计数问题。

第6章 关 系

在许多情况下集合的元素之间都存在某种关系。每天我们都要涉及各种关系，例如一个商行和它的电话号码之间的关系，雇员与其工资之间的关系，一个人与一个亲属之间的关系，等等。在数学中我们研究的关系，有如一个正整数与被它整除的一个正整数、一个整数与和它模5同余的一个整数、一个实数与一个比它大的实数之间的关系等等。在计算机科学中常常出现的关系，有如一个程序与它所使用的一个变量、一种计算机语言与这个语言的一个有效语句之间的关系等。

集合的元素之间的关系被表示成一种结构，这种结构叫做关系。可以用关系来求解问题，例如确定在一个网络中的哪两个城市之间开通航线，为一个复杂课题的不同阶段的工作找一个可行的次序，或者产生一个有用的方式以便在计算机数据库中存储信息。

6.1 关系及其性质

6.1.1 引言

可以用两个相关元素构成的有序对来表达两个集合的元素之间的关系，这是一种最直接的方式。为此，有序对的集合就叫做二元关系。在这一节，我们引入用于描述二元关系的基本术语。在本章的后面，我们将使用关系来求解涉及通信网络、项目调度以及识别集合中具有共同性质的元素等问题。

定义1 设 A 和 B 是集合，一个从 A 到 B 的二元关系是 $A\times B$ 的子集。

换句话说，一个从 A 到 B 的二元关系是有序对的集合 R，其中每个有序对的第一个元素取自 A 而第二个元素取自 B。用记号 aRb 表示 $(a,b)\in R$，$a\not R b$ 表示 $(a,b)\notin R$。当 (a,b) 属于 R 时叫做 a 与 b 有关系 R。

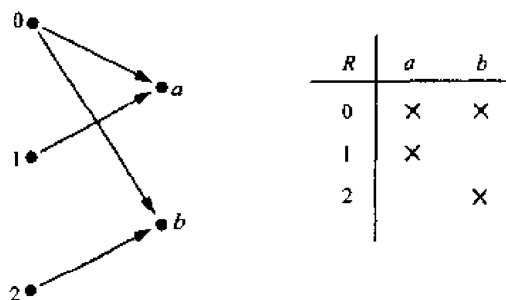
二元关系表示两个集合的元素之间的关系。在本章的后面将引入 n 元关系，它表示在三个以上集合中元素之间的关系。当不发生混淆时我们将省去"二元"这个词。

下面是关系的例子。

例1 设 A 是你们学校的学生的集合，B 是课程的集合。令 R 是由 (a,b) 对构成的关系，其中 a 是选修课程 b 的学生。例如，如果 Jason Goodfriend 和 Deborah Sherman 是选修 CS518，即离散数学，有序对（Jason Goodfriend，CS518）和（Deborah Sherman，CS518）属于 R。如果 Jason Goodfriend 也选修 CS510，即数据结构，那么有序对（Jason Goodfriend，CS510）也属于 R。但是，如果 Deborah Sherman 没有选修 CS510，那么有序对（Deborah Sherman，CS510）不在 R 中。 ■

例2 设 A 是所有城市的集合，B 是美国的50个州的集合。如下定义关系 R：如果 a 城市是在 b 州则 (a,b) 属于 R。例如，(Boulder，科罗拉多州)，(Bangor，缅因州)，(Ann Arbor，密执安州)，(Cupertino，加利福尼亚州)和(Red Bank，新泽西州)是在 R 中。 ■

例3 设 $A=\{0,1,2\}$，$B=\{a,b\}$，那么 $\{(0,a),(0,b),(1,a),(2,b)\}$ 是从 A 到 B 的关系。这意味着，比如说有 $0Ra$，但没有 $1Rb$。关系可以用图来表示，如图6-1所示，用箭头来表示有序对。另一种表示关系的方式就是用一张表，这也在图6-1中给出。在6.3节我们将更详细地讨论关系的表示。■



R	a	b
0	×	×
1	×	
2		×

图6-1 在例3的关系 R 中的有序对

6.1.2 函数作为关系

一个从集合 A 到集合 B 的函数 f（如1.6节的定义）对于 A 中的每个元素都指定 B 中一个唯一的元素。f 的图示是使得 $b=f(a)$ 的有序对 (a,b) 的集合。由于 f 的图示是 $A\times B$ 的子集，它就是一个从 A 到 B 的关系。此外，函数的图示有下述性质：A 的每个元素是图中恰好一个有序对的第一元素。

相反，如果 R 是从 A 到 B 的关系，并且使得 A 中的每个元素是 R 中恰好一个有序对的第一元素，那么 R 的图示就可以定义一个函数。只要对 A 的每个元素指定唯一的元素 $b\in B$ 使得 $(a,b)\in R$ 就可以做到。

可以用关系表达在集合 A 和集合 B 之间的一对多的相关性，其中 A 的一个元素可以与 B 中多个元素相关。函数表示了这样一种关系，对于 A 中的每个元素恰好只有一个 B 中的元素与之相关。

6.1.3 集合上的关系

集合 A 到它自身的关系是特别令人感兴趣的。

定义2 集合 A 上的关系是从 A 到 A 的关系。

换句话说，集合 A 上的关系是 $A\times A$ 的子集。

例4 设 A 是集合 $\{1,2,3,4\}$，A 上的关系 $R=\{(a,b)\mid a$ 整除 $b\}$ 中有哪些有序对？

解 (a,b) 在 R 中，当且仅当 a 和 b 是不超过4的正整数且 a 整除 b，我们看到

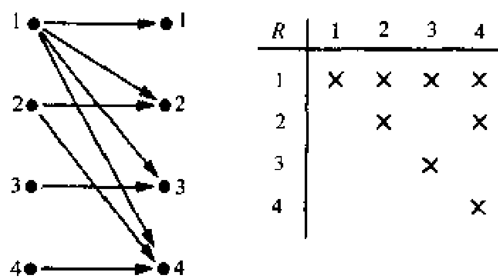
$$R=\{(1,1),(1,2),(1,3),(1,4),(2,2),(2,4),(3,3),(4,4)\}$$

在这个关系中的有序对的图和表的表示都在图6-2中给出。■

下面给出了某些整数集合上的关系的实例。

例5 考虑下面这些整数集合上的关系：

$$R_1=\{(a,b)\mid a\leqslant b\}$$
$$R_2=\{(a,b)\mid a>b\}$$
$$R_3=\{(a,b)\mid a=b \text{ 或 } a=-b\}$$
$$R_4=\{(a,b)\mid a=b\}$$
$$R_5=\{(a,b)\mid a=b+1\}$$



R	1	2	3	4
1	×	×	×	×
2		×		×
3			×	
4				×

图6-2 例4中关系 R 的有序对

$R_6 = \{(a,b) \mid a + b \leqslant 3\}$

其中哪些关系包含了有序对(1,1),(1,2),(2,1),(1,-1)以及(2,2)?

注意 和例1~4的关系不同，这些是无穷集合上的关系。

解 有序对(1,1)在 R_1，R_3，R_4 和 R_6 中；有序对(1,2)在 R_1 和 R_6 中；有序对(2,1)在 R_2，R_5 和 R_6 中；有序对(1,-1)在 R_2，R_3 和 R_6 中；最后，有序对(2,2)在 R_1，R_3 和 R_4 中。■

不难确定有穷集上的关系个数，因为集合 A 上的关系仅仅是 $A \times A$ 的子集。

例6 n 元素集合上有多少个关系?

解 集合 A 上的关系是 $A \times A$ 的子集。因为当 A 是 n 元素集合时 $A \times A$ 有 n^2 个元素，并且 m 个元素的集合有 2^m 个子集，故 $A \times A$ 的子集有 2^{n^2} 个。于是 n 元素集合有 2^{n^2} 个关系。■

6.1.4 关系的性质

有若干用于把集合上的关系分类的性质。这里我们只介绍其中最重要的性质。

在某些关系中一个元素总是与自己相关。例如，设 R 是所有的人的集合上的关系，若 x 和 y 有相同的母亲和相同的父亲，那么 (x,y) 属于 R。于是对于每个人 x，有 xRx。

定义3 如果对每个元素 $a \in A$ 有 $(a,a) \in R$，那么集合 A 上的关系 R 叫做自反的。

我们看到如果 A 的每个元素都和它自己相关，A 上的关系就是自反的。下面的例子说明了自反关系的概念。

例7 考虑{1,2,3,4}上的关系

$R_1 = \{(1,1)(1,2),(2,1),(2,2),(3,4),(4,1)(4,4)\}$

$R_2 = \{(1,1),(1,2),(2,1)\}$

$R_3 = \{(1,1),(1,2),(1,4),(2,1),(2,2),(3,3),(4,1),(4,4)\}$

$R_4 = \{(2,1),(3,1),(3,2),(4,1),(4,2),(4,3)\}$

$R_5 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(3,4),(4,4)\}$

$R_6 = \{(3,4)\}$

其中哪些是自反的?

解 关系 R_3 和 R_5 是自反的，因为它们都包含了所有形如 (a,a) 的对，即(1,1)、(2,2)、(3,3)和(4,4)。其他的关系不是自反的，因为它们不包含所有这些有序对。特别，R_1、R_2、R_4 和 R_6 不是自反的，因为(3,3)都不在这些关系里。■

例8 例5中哪些关系是自反的?

解 这个例子中的自反关系是 R_1（因为对每个整数 a 有 $a \leqslant a$），R_3 和 R_4。对于这个例子中其他的关系容易找到不在关系中的形如 (a,a) 的对。(留给读者练习。) ■

例9 正整数集合上的整除关系是自反的吗?

解　因为只要 a 是正整数就有 $a|a$，整除关系是自反的。■

在某些关系中一个元素与第二个元素相关，当且仅当第二个元素也与第一个元素相关。比如一个关系由 (x,y) 对构成，其中 x 和 y 是你校的学生并且他们至少学一门公共课程。这个关系就有这种性质。而某些关系有另一种性质，即如果一个元素与第二个元素相关，那么第二个元素就不与第一个元素相关。比如一个关系由 (x,y) 对构成，其中 x 和 y 是你校的学生，且 x 比 y 的平均成绩高。这个关系就有后一种性质。

定义 4　对于 a，$b\in A$，如果只要 $(a,b)\in R$ 就有 $(b,a)\in R$，则集合 A 上的关系 R 叫做对称的。对果对于 a，$b\in A$，仅当 $a=b$ 时 $(a,b)\in R$ 和 $(b,a)\in R$，则集合 A 上的关系 R 叫做反对称的。

就是说，关系 R 是对称的，当且仅当如果 a 与 b 相关则 b 与 a 就相关。关系 R 是反对称的，当且仅当不存在由不同元素 a 和 b 构成的有序对，使得 a 与 b 相关并且 b 与 a 也相关。对称与反对称的概念不是对立的，因为一个关系可以同时有这两种性质或者两种性质都没有（见节末的练习 6）。一个关系如果包含了某些形如 (a,b) 的对，其中 $a\neq b$，这个关系就不可能同时是对称和反对称的。

例 10　例 7 中的哪些关系是对称的？哪些是反对称的？

解　关系 R_2 和 R_3 是对称的，因为在这两个关系中只要 (a,b) 属于关系就有 (b,a) 属于关系。对于关系 R_2，唯一需要检查的就是 $(1,2)$ 和 $(2,1)$ 属于这个关系。对于 R_3，必须要检查 $(1,2)$ 和 $(2,1)$ 属于这个关系，还有 $(1,4)$ 和 $(4,1)$ 也属于这个关系。读者应该能够验证其他的关系中没有一个是对称的。只需找到一个有序对 (a,b) 在关系中，但 (b,a) 不在关系中。

R_4、R_5 和 R_6 都是反对称的。其中每一个关系都不存在由元素 a 和 b 构成的有序对，使得 $a\neq b$ 但 (a,b) 和 (b,a) 都属于这个关系。读者应该能验证其他关系中没有一个是反对称的。只需找到有序对 (a,b) 满足 $a\neq b$，而 (a,b) 和 (b,a) 都属于这个关系。■

例 11　例 5 中的哪些关系是对称的？哪些是反对称的？

解　关系 R_3、R_4 和 R_6 是对称的。R_3 是对称的，因为如果 $a=b$ 或 $a=-b$，那么就有 $b=a$ 或 $b=-a$。R_4 是对称的，因为由 $a=b$ 推出 $b=a$。R_6 是对称的，因为由 $a+b\leqslant 3$ 推出 $b+a\leqslant 3$。读者应该能验证其他关系没有一个是对称的。

关系 R_1，R_2，R_4 和 R_5 是反对称的。R_1 是反对称的，因为不等式 $a\leqslant b$ 和 $b\leqslant a$ 推出 $a=b$。R_2 是反对称的，因为 $a>b$ 和 $b>a$ 是不可能的。R_4 是反对称的，因为两个元素相对于 R_4 有关系，当且仅当它们是相等的。R_5 是反对称的，因为 $a=b+1$ 和 $b=a+1$ 是不可能的。读者应该能验证其他关系没有一个是反对称的。■

例 12　正整数集合上的整除关系是对称的吗？它是反对称的吗？

解　这个关系不是对称的，因为 $1|2$，但 $2\nmid 1$。它是反对称的，因若 a 和 b 是正整数且 $a|b$ 和 $b|a$，那么 $a=b$（这个验证留给读者作练习）。■

设 R 是有序对 (x,y) 构成的关系，其中 x 与 y 是你校的学生，且 x 比 y 得到更多的学分。假设 x 与 y 有关系并且 y 与 z 有关系，这意味着 x 比 y 得到更多的学分并且 y 与 z 得到

更多的学分。可以断言 x 比 z 得到更多的学分，因此 x 与 z 有关系。我们证明了 R 有传递性，这个性质定义如下。

定义 5　对于 a，b，$c \in A$，如果 $(a,b) \in R$，并且 $(b,c) \in R$，则 $(a,c) \in R$，那么集合 A 上的关系 R 叫做**传递的**。

例 13　例 7 的关系中哪些是传递的？

解　R_4，R_5 和 R_6 是传递的。对这些关系我们可以证明，若 (a,b) 和 (b,c) 属于一个关系，则 (a,c) 也属于这个关系由此验证它们是传递的。例如，R_4 是传递的，因为只有 $(3,2)$ 和 $(2,1)$，$(4,2)$ 和 $(2,1)$，$(4,3)$ 和 $(3,1)$ 以及 $(4,3)$ 和 $(3,2)$ 是这种有序对，而 $(3,1)$，$(4,1)$ 和 $(4,2)$ 属于 R_4。读者应该能验证 R_5 和 R_6 也是传递的。

R_1 不是传递的，因为 $(3,4)$ 和 $(4,1)$ 属于 R_1，但 $(3,1)$ 不属于 R_1。R_2 不是传递的，因为 $(2,1)$ 和 $(1,2)$ 属于 R_2，但 $(2,2)$ 不属于 R_2。R_3 不是传递的，因为 $(4,1)$ 和 $(1,2)$ 属于 R_3，但 $(4,2)$ 不属于 R_3。 ■

例 14　例 5 中的哪些关系是传递的？

解　关系 R_1、R_2、R_3 和 R_4 是传递的。R_1 是传递的，因为 $a \leqslant b$ 和 $b \leqslant c$ 推出 $a \leqslant c$。R_2 是传递的，因为 $a > b$ 和 $b > c$ 推出 $a > c$。R_3 是传递的，因为 $a = \pm b$ 和 $b = \pm c$ 推出 $a = \pm c$。正如读者应该验证的，显然 R_4 也是传递的。R_5 不是传递的，因为 $(2,1)$ 和 $(1,0)$ 属于 R_5，但 $(2,0)$ 不属于 R_5。R_6 不是传递的，因为 $(2,1)$ 和 $(1,2)$ 属于 R_6，但 $(2,2)$ 不属于 R_6。 ■

例 15　正整数集合上的“整除”关系是传递的吗？

解　假设 a 整除 b 且 b 整除 c，那么存在正整数 k 和 l 使得 $b = ak$ 和 $c = bl$。因此 $c = akl$，即 a 整除 c。从而证明了这个关系是传递的。 ■

下面的例子显示了怎样计数具有特殊性质的关系的个数。

例 16　n 元素集合上有多少个自反的关系？

解　A 上的关系是 $A \times A$ 的子集。因此，要通过指定 n^2 个有序对中的每一个是否在 R 中来确定关系。但是，如果 R 是自反的，对于 $a \in A$，n 个有序对 (a,a) 中的每一个都必须在 R 中。其他 $n(n-1)$ 个形如 (a,b) 的有序对，$a \neq b$，可能在也可能不在 R 中。因此，由计数的乘积法则，存在 $2^{n(n-1)}$ 个自反的关系［这就是选择具有 $a \neq b$ 的每个元素 (a,b) 是否属于 R 的方式数］。 ■

n 元素集合上的对称关系和反对称关系数可以用和例 16 类似的推理得出（见本节末的练习 25）。计数 n 元素集合上的传递关系数的问题已超出本书的范围。

6.1.5　关系的组合

因为从 A 到 B 的关系是 $A \times B$ 的子集，可以按照两个集合组合的任何方式来组合两个从 A 到 B 的关系。考虑下面的例子。

例 17　设 $A = \{1,2,3\}$ 和 $B = \{1,2,3,4\}$。组合关系 $R_1 = \{(1,1),(2,2),(3,3)\}$ 和

$R_2=\{(1,1),(1,2),(1,3),(1,4)\}$可以得到

$$R_1\cup R_2=\{(1,1),(1,2),(1,3),(1,4),(2,2),(3,3)\}$$
$$R_1\cap R_2=\{(1,1)\}$$
$$R_1-R_2=\{(2,2),(3,3)\}$$
$$R_2-R_1=\{(1,2),(1,3),(1,4)\}$$

■

例 18 设 A 和 B 分别是学校的所有学生和所有课程的集合。假设 R_1 由所有有序对 (a,b) 组成，其中 a 是选修课程 b 的学生。R_2 由所有的有序对 (a,b) 构成，其中课程 b 是 a 的必修课。什么是关系 $R_1\cup R_2$，$R_1\cap R_2$，$R_1\oplus R_2$，R_1-R_2 和 R_2-R_1?

解 关系 $R_1\cup R_2$ 由所有的有序对 (a,b) 组成，其中 a 是一个学生，他或者选修了课程 b，或者课程 b 是他的必修课。$R_1\cap R_2$ 是所有有序对 (a,b) 的集合，其中 a 是一个学生，他选修了课程 b 并且课程 b 也是 a 的必修课。$R_1\oplus R_2$ 由所有的有序对 (a,b) 组成，其中学生 a 已经选修了课程 b，但课程 b 不是 a 的必修课，或者课程 b 是 a 的必修课，但是 a 没有选修它。R_1-R_2 是所有有序对 (a,b) 的集合，其中 a 已经选修了课程 b，但 b 不是 a 的必修课。R_2-R_1 是所有有序对 (a,b) 的集合，其中 b 是 a 的必修课，但 a 没有选它。 ■

组合关系有另一种方式，这种方式和函数的合成相似。

定义 6 设 R 是从集合 A 到集合 B 的关系，S 是从集合 B 到集合 C 的关系。R 和 S 的合成是由有序对 (a,c) 构成的关系，其中 $a\in A$、$c\in C$，并且对于它们存在一个元素 $b\in B$，使得 $(a,b)\in R$ 和 $(b,c)\in S$。我们用 $S\circ R$ 表示 R 与 S 的合成。

下面的例子说明了关系的合成是怎样构成的。

例 19 R 是从 $\{1,2,3\}$ 到 $\{1,2,3,4\}$ 的关系且 $R=\{(1,1),(1,4),(2,3),(3,1),(3,4)\}$，$S$ 是从 $\{(1,2,3,4)\}$ 到 $\{0,1,2\}$ 的关系且 $S=\{(1,0),(2,0),(3,1),(3,2),(4,1)\}$，$R$ 与 S 的合成是什么?

解 $S\circ R$ 是由所有 R 中的有序对和 S 中有的序对构成的，其中在 R 中有序对的第二元素与 S 中有序对的第一元素相同。例如，R 中的有序对 $(2,3)$ 和 S 中的有序对 $(3,1)$ 产生了 $S\circ R$ 中的有序对 $(2,1)$。计算所有在 $R\circ S$ 中的有序对，我们得到

$$S\circ R=\{(1,0),(1,1),(2,1),(2,2),(3,0),(3,1)\}$$

■

可以递归地使用两个关系的合成来定义关系 R 的幂。

定义 7 设 R 是集合 A 上的关系。幂 R^n，$n=1$，2，3，…，递归地定义为

$$R^1=R \text{ 和 } R^{n+1}=R^n\circ R$$

这个定义证明了 $R^2=R\circ R$，$R^3=R^2\circ R=(R\circ R)\circ R$，等等。

例 20 设 $R=\{(1,1),(2,1),(3,2),(4,3)\}$。求幂 R^n，$n=2$，3，4…。

解 因为 $R^2=R\circ R$，我们得到 $R^2=\{(1,1),(2,1),(3,1),(4,2)\}$。进一步，因为 $R^3=R^2\circ R$，$R^3=\{(1,1),(2,1),(3,1),(4,1)\}$。其他的计算证明了 R^4 和 R^3 一样，因此 $R^4=\{(1,1),(2,1),(3,1),(4,1)\}$。从而对 $n=5$，6，7…，有 $R^n=R^3$。读者应该能够验证这

个结果。■

下面的定理证明一个传递关系是幂是该关系的子集。6.4 节将要用到这一结果。

定理 1　集合 A 上的关系 R 是传递的，当且仅对 $n=1, 2, 3$，有 $R^n \subseteq R$。

证　首先证明定理的充分条件。假设对 $n=1, 2, 3$，有 $R^n \subseteq R$。特别地，有 $R^2 \subseteq R$。为证明 R 的传递性，注意到如果 $(a,b) \in R$ 并且 $(b,c) \in R$，根据合成定义就有 $(a,c) \in R^2$。因为 $R^2 \subseteq R$，这就意味着 $(a,c) \in R$。因此 R 是传递的。

我们将使用数学归纳法证明定理的必要条件。对于 $n=1$，定理的这个结果是显而易见的。

假定 $R^n \subseteq R$，其中 n 是一个正整数，这是归纳假设。为完成归纳步骤，必须证明这将推出 R^{n+1} 也是 R 的子集。为证明这一点，假设 $(a,b) \in R^{n+1}$，那么因为 $R^{n+1} = R^n \circ R$，存在元素 $x \in A$，使得 $(a,x) \in R$ 并且 $(x,b) \in R^n$。由归纳假设，即 $R^n \subseteq R$，推出 $(x,b) \in R$。下一步，因为 R 是传递的，以及 $(a,x) \in R$ 和 $(x,b) \in R$，得到 $(a,b) \in R$。这就证明了 $R^{n+1} \subseteq R$，从而完成了证明。□

练习

1. 列出从 $A=\{0,1,2,3,4\}$ 到 $B=\{0,1,2,3\}$ 的关系 R 中的有序对，其中 $(a,b) \in R$，当且仅当
   a) $a=b$　　b) $a+b=4$
   c) $a>b$　　d) $a \mid b$
   e) $\gcd(a,b)=1$　　f) $\operatorname{lcm}(a,b)=2$
2. a) 列出集合 $\{1,2,3,4,5,6\}$ 上的关系 $R=\{(a,b) \mid a \text{ 整除 } b\}$ 中所有的有序对。
   b) 仿照例 4 用图表示这个关系。
   c) 仿照例 4 用表表示这个关系。
3. 对集合 $\{1,2,3,4\}$ 上的下面每一个关系确定其是否是自反的；是否是对称的；是否是反对称的；是否是传递的。
   a) $\{(2,2),(2,3),(2,4),(3,2),(3,3),(3,4)\}$
   b) $\{(1,1),(1,2),(2,1),(2,2),(3,3),(4,4)\}$
   c) $\{(2,4),(4,2)\}$
   d) $\{(1,2),(2,3),(3,4)\}$
   e) $\{(1,1),(2,2),(3,3),(4,4)\}$
   f) $\{(1,3),(1,4),(2,3),(2,4),(3,1),(3,4)\}$
4. 确定所有人的集合上的关系 R 是否是自反的、对称的、反对称的和传递的，其中 $(a,b) \in R$，当且仅当
   a) a 比 b 高。　　b) a 和 b 生在同一天。
   c) a 和 b 的名字相同。　　d) a 和 b 有共同的祖父母。
5. 确定所有整数集合上的关系 R 是否是自反的、对称的、反对称的和传递的，其中 $(x,y) \in R$，当且仅当
   a) $x \neq y$　　b) $xy \geq 1$　　c) $x=y+1$ 或 $x=y-1$

d) $x \equiv y \pmod 7$　　e) x 是 y 的倍数　　f) x 与 y 都是负的或都是非负的

g) $x = y^2$　　h) $x \geq y^2$

6. 给出一个集合上的关系的例子，要求它是

a) 对称的和反对称的。　　b) 既不是对称的也不是反对称的。

☞如果对于每个 $a \in A$，有 $(a,a) \notin R$，那么集合 A 上的关系 R 是**反自反的**。即如果没有 A 中的元素与自己有关系，关系 R 就是反自反的。

7. 练习 3 的哪些关系是反自反的？
8. 练习 4 的哪些关系是反自反的？
9. 集合上的关系可能既不是自反的也不是反自反的吗？

一个关系 R 叫做**非对称的**，如果由 $(a,b) \in R$ 推出 $(b,a) \in R$。

10. 练习 3 的哪些关系是非对称的？
11. 练习 4 的哪些关系是非对称的？
12. 非对称的关系也一定是反对称的吗？反对称的关系也一定是非对称的吗？对你的答案说明理由。
13. 从 m 元素集合到 n 元素集合有多少个不同的关系？

☞设 R 是从集合 A 到集合 B 的关系。从 B 到 A 的**逆关系**是有序对的集合 $\{(b,a) \mid (a,b) \in R\}$，记作 R^{-1}，**补关系** $\overline{R}$ 是有序对的集合 $\{(a,b) \mid (a,b) \notin R\}$。

14. 设 R 是整数集合上的关系 $R = \{a,b \mid a < b\}$。求

a) R^{-1}　　b) $\overline{R}$

15. 设 R 是正整数集合上的关系 $R = \{(a,b) \mid a$ 整除 $b\}$。求

a) R^{-1}　　b) $\overline{R}$

16. 设 R 是美国所有州的集合上的关系，R 由有序对 (a,b) 构成，其中 a 州与 b 州相邻接。求

a) R^{-1}　　b) $\overline{R}$

17. 设从 A 到 B 的函数 f 是一一对应的。令 R 是和 f 的图相等的关系，即 $R = \{(a, f(a)) \mid a \in A\}$。什么是逆关系 R^{-1}？
18. 令 $R_1 = \{(1,2),(2,3),(3,4)\}$，$R_2 = \{(1,1),(1,2),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3),(3,4)\}$ 是从 $\{1,2,3\}$ 到 $\{1,2,3,4\}$ 的关系。求

a) $R_1 \cup R_2$　　b) $R_1 \cap R_2$，　　c) $R_1 - R_2$　　d) $R_2 - R_1$

19. 设 A 是你校的学生的集合，B 是学校图书馆的书的集合。令 R_1 和 R_2 是由所有有序对 (a,b) 组成的关系，在 R_1 中，学生 a 在一门课程中需要读书 b，在 R_2 中，学生 a 已经读过书 b。描述在下面每个关系中的有序对。

a) $R_1 \cup R_2$　　b) $R_1 \cap R_2$　　c) $R_1 \oplus R_2$　　d) $R_1 - R_2$　　e) $R_2 - R_1$

20. 设 R 是关系 $\{(1,2),(1,3),(2,3),(2,4),(3,1)\}$，$S$ 是关系 $\{(2,1),(3,1),(3,2),(4,2)\}$。求 $S \circ R$。
21. 设 R 是由人的集合上的有序对 (a,b) 组成的关系，其中 a 是 b 的父母。设 S 是由人的

集合上的有序对(a,b)组成的关系，其中a是b的兄弟姐妹。$S \circ R$和$R \circ S$是什么关系？

22. 列出集合$\{0,1\}$上的16个不同的关系。
23. $\{0,1\}$上的16个不同的关系中有多少个包含了对$(0,1)$？
24. 你在练习22列出的16个关系中哪些是

    a) 自反的？　b) 反自反的？　c) 对称的？

    d) 反对称的？　e) 非对称的？　f) 传递的？

*25. n元素集合上有多少个关系是

    a) 对称的？　b) 反对称的？

    c) 非对称的？　d) 反自反的？

    e) 自反的和对称的？　f) 既不是自反的也不是反自反的？

*26. n元素集合上有多少个传递的关系？如果

    a) $n=1$　b) $n=2$　c) $n=3$

27. 找出在下面定理证明中的错误。

    "定理"：设R是集合A上的对称和传递的关系，则R是自反的。

    "证明"：设$a \in A$。取元素$b \in A$使得$(a,b) \in R$。由于R是对称的，因而有$(b,a) \in R$。

    现在使用传递性，由$(a,b) \in R$和$(b,a) \in R$可以断言$(a,a) \in R$。

28. 假设R和S是集合A上自反的关系。证明或反证下面的论断。

    a) $R \cup S$是自反的。

    b) $R \cap S$是自反的。

    c) $R \oplus S$是反自反的。

    d) $R-S$是自反的。

    e) $S \circ R$是自反的。

29. 证明集合A上的关系R是对称的，当且仅当$R=R^{-1}$，其中R^{-1}是R的逆关系。
30. 证明集合A上的关系R是反对称的，当且仅当$R \cap R^{-1}$是对角线关系$\Delta=\{(a,a) \mid a \in A\}$的子集。
31. 证明集合A上的关系R是自反的，当且仅当逆关系R^{-1}是自反的。
32. 证明集合A上的关系R是自反的，当且仅当补关系$\overline{R}$是反自反的。
33. 设R是自反的和传递的关系，证明对所有的正整数n，$R^n=R$。
34. 设R是集合$\{1,2,3,4,5\}$上的关系，R包含有序对$(1,1)$，$(1,2)$，$(1,3)$，$(2,3)$，$(2,4)$，$(3,1)$，$(3,4)$，$(3,5)$，$(4,2)$，$(4,5)$，$(5,1)$，$(5,2)$和$(5,4)$。求

    a) R^2　b) R^3　c) R^4　d) R^5

35. 设R是集合A上的自反关系，证明对所有的正整数n，R^n也是自反的。

*36. 设R是对称关系，证明对所有的正整数n，R^n也是对称的。

37. 假设关系R是反自反的，R^2一定是反自反的吗？对你的答案给出理由。

6.2 n 元关系及其应用

6.2.1 引言

在两个以上集合的元素中常常会产生某种关系。例如，存在学生的姓名、学生的专业以及学生的平均绩分点之间的关系。类似地，一个航班的航空公司、航班号、出发地、目的地、起飞时间和到达时间等也有一种关系。在数学中也有这种关系。例如有3个整数，其中第一个整数比第二个整数大，而第二个整数比第三个整数大。另一个例子是直线上的点之间的关系，即当第二个点在第一和第三个点之间时，这三个点有关系。

本节我们将研究两个以上集合的元素之间的关系。这种关系叫 n 元关系。可以用这种关系表示计算机的数据库。这种表示在我们回答对数据库中所存信息的查询时提供帮助，例如：哪个航班在午夜3点到4点之间降落在O'Hare机场？你们学校的哪些二年级学生是主修数学或计算机科学并且平均绩分点大于3.0？公司的哪些雇员为这个公司工作不到5年但报酬超过50 000美元？

6.2.2 n 元关系

我们从定义开始。

定义1 设 A_1，$A_2\cdots$，A_n 是集合。在这些集合上的 n 元关系是 $A_1\times A_2\times\cdots\times A_n$ 的子集。这些集合 A_1，A_2，$\cdots A_n$ 叫做关系的域，n 叫做它的阶。

例1 设 R 是由三元组 (a,b,c) 构成的关系，其中 a、b、c 是满足 $a<b<c$ 的整数。那么 $(1,2,3)\in R$，但 $(2,4,3)\notin R$。这个关系的阶是3。它的域都等于整数集合。 ■

例2 设 R 是由5元组 (A,N,S,D,T) 构成的表示飞机航班的关系，其中 A 是航空公司，N 是航班号，S 是出发地，D 是目的地，T 是起飞时间。例如，如果Nadir直达航空公司在15:00有从Newark到Angor的963航班，那么(Nadir，963，Newark，Bangor，15:00)属于R。这个关系的阶是5，它的域是所有航空公司的集合，航班号的集合，城市的集合，以及时间的集合。 ■

6.2.3 数据库和关系

数据库信息操作所需要的时间依赖于这些信息是怎样存储的。插入和删除记录，更新记录，检索记录以及从一些重叠的数据库中组合记录的操作，在一个大型数据库中每天要执行几百万次。由于这些操作的重要性，已经开发了数据库表示的各种方法。我们将讨论其中的一种基于关系概念的方法，叫做*关系数据模型*。

数据库由记录组成，这些记录是由字段构成的n元组。这些字段是n元组的数据项。例如，学生记录的数据库可以由包含学生的姓名、学号、专业、平均成绩（GPA）的字段构成。关系数据模型把一个记录的数据库表示成一个n元关系。于是，学生记录可以被表示成形如（学生姓名、学号、专业、GPA）的4元组。6个记录的一个数据库样本是：

(Ackermann，231455，计算机科学，3.88)

(Adams，888323，物理学，3.45)

(Chou，102147，计算机科学，3.79)

(Goodfriend，453876，数学，3.45)

(Rao，678543，数学，3.90)

(Stevens，786576，心理学，2.99)

用于表示数据库的关系叫做表，因为这些关系常常用表来给出。例如，同样的学生数据库给在表 6-1 中。

当 n 元组的某个域的值能够确定这个 n 元组时，n 元关系的这个域就叫做**主键码**。这就是说当关系中没有两个 n 元组在这个域有相同的值时这个域就是主键码。

常常要从数据库增加或删除记录。由于这一点，一个域是主键码的性质是随时间而改变的。所以，一个主键码应该选择那种无论数据库怎样改变都能继续存在的字段。用数据库的内涵的主键码就可以做到这一点，它包含了在表示这个数据库的 n 元关系中所有可能含有的 n 元组。

例 3 假设将来不再增加 n 元组，对于表 6-1 所示的 n 元关系，哪个域是主键码？

表 6-1

学生姓名	学号	专业	GPA
Ackermann	231455	计算机科学	3.88
Adams	888323	物理学	3.45
Chou	102147	计算机科学	3.79
Goodfriend	453876	数学	3.45
Rao	678543	数学	3.90
Stevens	786576	哲学	2.99

解 因为在这个表中每个学生的姓名只有一个 4 元组，学生姓名的域是主键码。类似地，在这个表中学号是唯一的，学号的域也是主键码。但是，所学专业的域不是主键码，因为多于一个 4 元组包含同样的专业。平均成绩的域也不是主键码，因为有 2 个 4 元组包含了同样的 GPA。(哪 2 个?) ■

在一个 n 元关系中域的组合也可以唯一地标识 n 元组。当一组域的值确定了一个关系中的 n 元组时，这些域的笛卡儿积就叫做**复合键码**。

例 4 对于表 6-1 中的 n 元关系，假设不增加 n 元组，所学专业的域与 GPA 的域的笛卡儿积是复合键码吗？

解 这个表中没有两个 4 元组同时有同样的专业和同样的 GPA，因此这个笛卡儿积是一个复合键码。 ■

因为主键码和复合键码用于唯一的标识数据库中的记录，当新的记录被加到这个数据库时键码要保持有效是非常重要的。因此，应该做检测以保证在这个或这些相应的字段中每个新记录与表中所有其他的记录不同。例如，使用学号作为学生记录的键码是有意义的，因为没有两个学生有同样的学号。一个大学不应该使用姓名字段作为键码，因为有可能两个学生有同样的姓名（如 John Smith）。

可以用各种 n 元关系上的运算构造新的 n 元关系。这里要讨论两个运算，即投影和连

接运算。使用投影运算通过删除关系的每个记录中同样的一些字段来构成新的 n 元关系。

定义 2 投影 $P_{i_1,i_2,\cdots,i_m}$ 将 n 元组 $(a_1,a_2,\cdots,a_n)$ 映到 m 元组 $(a_{i_1},a_{i_2},\cdots,a_{i_m})$，其中 $m\leqslant n$。

换句话说，投影 $P_{i_1,i_2,\cdots,i_m}$ 删除了 n 元组的 $n-m$ 个分量，保留了第 i_1，i_2，…，i_m 分量。

例 5 当投影 $P_{1,3}$ 施用到 4 元组 (2,3,0,4)，(Jane Doe,234111001,地理学,3.14) 以及 (a_1,a_2,a_3,a_4) 时结果是什么？

解 $P_{1,3}$ 把这些 4 元组分别映到 (2,0)，(Jane Doe,地理学) 和 (a_1,a_3)。 ■

下面的例子说明了怎样使用投影来产生新关系。

例 6 当投影 $P_{1,4}$ 施用于表 6-1 中的关系时结果是什么？

解 当施用投影 $P_{1,4}$ 时，表的第二和第三列被删除，得到了表示学生姓名和平均成绩的对。表 6-2 给出了这个投影的结果。 ■

当一个投影被施用到一个关系的表中有可能使行变少。当关系中的某些 n 元组在投影的 m 个分量中的每个分量的值都相同，而只在被投影删除的分量有不同的值时，就会出现这种情况。例如，考虑下面的例子。

例 7 当投影 $P_{1,2}$ 施用到表 6-3 的关系时得到什么表？

解 表 6-4 给出了当投影 $P_{1,2}$ 施用到表 6-3 时得到的关系。注意在施用了这个投影后行数减少。 ■

当两个表分享着相同的字段时，连接运算将这两个表组合成一个表。例如，一个表包含了航空公司、航班号和通道的字段，另一个表包含了航班号、通道和起飞时间的字段。可以将这两个表组合成一个包含航空公司、航班号、通道和起飞时间字段的表。

定义 3 设 R 是 m 阶关系且 S 是 n 阶关系。连接 $J_p(R,S)$ 是 $m+n-p$ 阶关系，其中 $p\leqslant m$ 和 $p\leqslant n$，它包含了所有的 $(m+n-p)$ 元组 $(a_1,a_2,\cdots,a_{m-p},c_1,c_2,\cdots,c_p,b_1,b_2,\cdots,b_{n-p})$，其中 m 元组 $(a_1,a_2,\cdots,a_{m-p},c_1,c_2,\cdots,c_p)$ 属于 R 且 n 元组 $(c_1,c_2,\cdots,c_p,b_1,b_2,\cdots,b_{n-p})$ 属于 S。

换句话说，连接运算 J_p 从两个关系产生一个新的关系，它把第一个关系的所有 m 元组和第二个关系的所有 n 元组组合起来，其中 m 元组的后 p 个分量与 n 元组的前 p 个分量相同。

表 6-2

学生姓名	GPA	学生姓名	GPA
Ackermann	3.88	Goodfriend	3.45
Adams	3.45	Rao	3.90
Chou	3.79	Stevens	2.99

表　6-3

学　生	专　　业	课程号	学　生	专　　业	课程号
Glauser	生物学	B1209	Marcus	数学	MS603
Glauser	生物学	MS475	Marcus	数学	CS322
Glauser	生物学	PY410	Miller	计算机科学	MS575
Marcus	数学	MS511	Miller	计算机科学	CS455

表　6-4

学　生	专　　业	学　生	专　　业	学　生	专　　业
Glauser	生物学	Marcus	数学	Miller	计算机科学

例 8　当用连接运算 J_2 组合表 6-5 和表 6-6 中的关系时所得结果关系是什么？

解　J_2 产生的关系给在表 6-7 中。 ■

从已知关系产生新关系的运算除了投影和连接运算以外还有其他运算。对这些运算的描述可以在关于数据库理论的书中找到。

表　6-5

教　授	系	课　号	教　授	系	课　号
Cruz	动物学	335	Grammer	物理学	544
Cruz	动物学	412	Grammer	物理学	551
Farber	心理学	501	Rosen	计算机科学	518
Farber	心理学	617	Rosen	数学	575

表　6-6

系	课　　号	教　　室	时　间
计算机科学	518	N521	2:00 P.M.
数学	575	N502	3:00 P.M.
数学	611	N521	4:00 P.M.
物理学	544	B505	4:00 P.M.
心理学	501	A100	3:00 P.M.
心理学	617	A110	11:00 A.M.
动物学	335	A100	9:00 A.M.
动物学	412	A100	8:00 A.M.

表　6-7

教　　授	系	课　　号	教　　室	时　间
Cruz	动物学	335	A100	9:00　A.M.
Cruz	动物学	412	A100	8:00　A.M.
Farber	心理学	501	A100	3:00　P.M.
Farber	心理学	617	A110	11:00 A.M.
Grammer	物理学	544	B505	4:00　P.M.
Rosen	计算机科学	518	N521	2:00　P.M.
Rosen	数学	575	N502	3:00　P.M.

表　6-8

航空公司	航　班	通　　道	目的地	起飞时间
Nadir	122	34	底特律	08:10
Acme	221	22	丹佛	08:17
Acme	122	33	安克雷奇	08:22
Acme	323	34	檀香山	08:30
Nadir	199	13	底特律	08:47
Acme	222	22	丹佛	09:10
Nadir	322	34	底特律	09:44

练习

1. 列出关系$\{(a,b,c)\mid a$，b 和 c 是整数且 $0<a<b<c<5\}$中的 3 元组。
2. 在关系$\{(a,b,c,d)\mid a,b,c,d$ 是正整数且 $abcd=6\}$中有哪些 4 元组？
3. 列出表 6-8 的关系中的 5 元组。
4. 假设不增加新的 n 元组，对下面表中的关系找出所有的主键码。
   a) 表 6-3　b) 表 6-5　c) 表 6-6　d) 表 6-8
5. 假设不增加新的 n 元组，对于表 6-8 中的数据库找出一个由两个字段构成的组合键码，其中一个字段是航空公司。
6. 当施用投影 $P_{2,3,5}$到 5 元组(a,b,c,d,e)时你能得到什么？
7. 哪个投影映射用于删除一个 6 元组的第一、第二和第四个分量？
8. 给出施用投影 $P_{1,2,4}$到表 6-8 以后得到的表。
9. 给出施用投影 $P_{1,4}$到表 6-8 以后得到的表。
10. 把连接运算 J_3 用到 5 元组的表和 8 元组的表后所得表中的 n 元组里有多少个分量？
11. 构造把连接运算 J_2 用到表 6-9 和表 6-10 的关系中所得到的表。

表　6-9

供货商	零件号	项目	供货商	零件号	项目
23	1092	1	31	3477	2
23	1101	3	32	6984	4
23	9048	4	32	9191	2
31	4975	3	33	1001	1

表　6-10

零件号	项目	数量	颜色代码	零件号	项目	数量	颜色代码
1001	1	14	8	4975	3	6	2
1092	1	2	2	6984	4	10	1
1101	3	1	1	9048	4	12	2
3477	2	25	2	9191	2	80	4

6.3 关系的表示

6.3.1 引言

有多种方法表示有穷集之间的关系。正如我们已经看到的，一种方法就是列出它的有序对。本节我们将讨论另外两种表示关系的方法。一种方法就是使用 0-1 矩阵。另一种方法就是使用有向图。

6.3.2 用矩阵表示关系

可以用 0-1 矩阵来表示一个有穷集之间的关系。假设 R 是从 $A=\{a_1,a_2,\cdots,a_m\}$ 到 $B=\{b_1,b_2,\cdots,b_n\}$ 的关系。(这里集合 A 和集合 B 的元素已经按照某一特定的但是任意的次序列出。此外，当 $A=B$ 时对于 A 和 B 使用同样的排序。) 关系 R 可以用矩阵 $M_R=[m_{ij}]$ 来表示，其中

$$m_{ij}=\begin{cases}1, & 如果(a_i,b_j)\in R\\ 0, & 如果(a_i,b_j)\notin R\end{cases}$$

换句话说，当 a_i 和 b_j 有关系时，表示 R 的 0-1 矩阵的 (i,j) 项是 1，当 a_i 和 b_j 没关系时，在这个位置是 0。(这种表示依赖于对 A 和 B 使用的排序。)

下面的例子说明用矩阵来表示关系。

例 1 假设 $A=\{1,2,3\}$，$B=\{1,2\}$。令 R 是从 A 到 B 的关系，且如果 $a\in A$、$b\in B$、$a>b$，则 R 包含 (a,b)。如果 $a_1=1$、$a_2=2$、$a_3=3$、$b_1=1$、$b_2=2$，表示 R 的矩阵是什么？

解 因为 $R=\{(2,1),(3,1),(3,2)\}$，关于 R 的矩阵是

$$\mathbf{M}_R=\begin{bmatrix}0 & 0\\ 1 & 0\\ 1 & 1\end{bmatrix}$$

$\mathbf{M}_R$ 中的 1 说明了有序对 (2,1)、(3,1) 和 (3,2) 属于 R，0 说明了没有其他的有序对属于 R。 ■

例 2 设 $A=\{a_1,a_2,a_3\}$，$b=\{b_1,b_2,b_3,b_4,b_5\}$。哪些有序对在下面的矩阵所表示的关系 R 中？

$$\mathbf{M}_R=\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 1 & 0\\ 1 & 0 & 1 & 0 & 1\end{bmatrix}$$

解 因为 R 由使得 $m_{ij}=1$ 的有序对 (a_i,b_j) 构成，因而

$$R=\{(a_1,b_2),(a_2,b_1),(a_2,b_3),(a_2,b_4),(a_3,b_1),(a_3,b_3),(a_3,b_5)\}$$ ■

集合上的关系的矩阵是一个方阵，可以用这个矩阵确定关系是否有某种性质。A 上的关系是自反的，如果对于每个 $a\in A$ 有 $(a,a)\in R$。于是，R 是自反的，当且仅当对 $i=1$，

2，…，n，$(a_i,a_i)\in R$。于是，R 是自反的当且仅当对 $i=1$，2，…，n，$m_{ii}=1$。换句话说，如果如图6-3所示，$\mathbf{M}_R$ 的主对角线的所有元素都等于1，那么 R 是自反的。

$$\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

图6-3 关于自反关系的0-1矩阵

如果 $(a,b)\in R$ 推出 $(b,a)\in R$，那么关系 R 是对称的。因此，集合 $A=\{a_1,a_2,\cdots,a_n\}$ 上的关系 R 是对称的，当且仅当只要 $(a_i,a_j)\in R$ 就有 $(a_j,a_i)\in R$。用矩阵 $\mathbf{M}_R$ 的术语来说，R 是对称的，当且仅当只要 $m_{ij}=1$ 就有 $m_{ji}=1$。这也意味着只要 $m_{ij}=0$，就有 $m_{ji}=0$，因此 R 是对称的，当且仅当对所有的整数对 i，j，其中 $i=1$，2…，n，$j=1$，2，…，n 都有 $m_{ij}=m_{ji}$。回顾2.6节矩阵转置的定义，我们看到 R 是对称的，当且仅当

$$\mathbf{M}_R=(\mathbf{M}_R)^t$$

即 $\mathbf{M}_R$ 是对称矩阵。对称关系的矩阵形式显示在图6-4a。

关系 R 是反对称的，当且仅当 $(a,b)\in R$ 和 $(b,a)\in R$ 推出 $a=b$。因此，反对称关系的矩阵有下述性质，即如果 $m_{ij}=1$，$i\neq j$，则 $m_{ji}=0$。或者换句话说，当 $i\neq j$ 时，$m_{ij}=0$ 或 $m_{ji}=0$。反对称关系的矩阵形式显示在图6-4b。

a) 对称的

b) 反对称的

图6-4 关于对称和反对称关系的0-1矩阵

例3 假设集合上的关系 R 由矩阵

$$\mathbf{M}_R=\begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 1\end{bmatrix}$$

表示。R 是自反的、对称的和反对称的吗？

解 因为个矩阵所有的对角线元素都等于1，R 是自反的。又由于 $\mathbf{M}_R$ 是对称的，因此 R 是对称的。也容易看出 R 不是反对称的。■

可以用布尔运算加和乘（在2.6节讨论）找两个关系的并和交的矩阵表示。假设集合 A 上的关系 R_1 和 R_2 分别由矩阵 $\mathbf{M}_{R_1}$ 和 $\mathbf{M}_{R_2}$ 表示。这些关系的并的矩阵表示在 $\mathbf{M}_{R_1}$ 或 $\mathbf{M}_{R_2}$ 为1的位置有1。这些关系的交的矩阵表示在 $\mathbf{M}_{R_1}$ 和 $\mathbf{M}_{R_2}$ 同时为1的位置有1。于是，这些关系的并和交的矩阵表示是

$$\mathbf{M}_{R_1\cup R_2}=\mathbf{M}_{R_1}\vee\mathbf{M}_{R_2}\quad 和\ \mathbf{M}_{R_1\cap R_2}=\mathbf{M}_{R_1}\wedge\mathbf{M}_{R_2}$$

例4 假设集合 A 上的关系 R_1 和 R_2 由矩阵

$$\mathbf{M}_{R_1}=\begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\ 和\ \mathbf{M}_{R_2}=\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 1 & 0 & 0\end{bmatrix}$$

表示。什么是 $R_1\cup R_2$ 和 $R_1\cap R_2$ 的矩阵表示？

解 这两个关系的矩阵是

$$\mathbf{M}_{R_1\cup R_2}=\mathbf{M}_{R_1}\vee M_{R_2}=\begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 1\\ 1 & 1 & 0\end{bmatrix}$$

$$\mathbf{M}_{R_1 \cap R_2} = \mathbf{M}_{R_1} \wedge M_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$ ■

现在我们来考虑怎样确定关系合成的矩阵。这个矩阵可以通过关系矩阵的布尔积（见2.6节）得到。特别地，假设 R 是从 A 到 B 的关系且 S 是从 B 到 C 的关系。又假设 A、B 和 C 分别有 m、n 和 p 个元素。令关于 $R \circ S$，R 和 S 的 0-1 矩阵分别为 $\mathbf{M}_{S \circ R} = [t_{ij}]$，$\mathbf{M}_R = [r_{ij}]$，$\mathbf{M}_s = [s_{ij}]$（这些矩阵的的大小为 $m \times p$，$m \times n$ 和 $n \times p$）。有序对 (a_i, c_j) 属于 $S \circ R$，当且仅当存在元素 b_k，使得 (a_i, b_k) 属于 R 并且 (b_k, c_j) 属于 S。从而，$t_{ij} = 1$，当且仅当对某个 k 有 $r_{ik} = s_{kj} = 1$。由布尔积的定义，这意味着

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot M_S$$

例 5　找出表示关系 $S \circ R$ 的矩阵，其中表示 R 和 S 的矩阵是

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad 和 \quad \mathbf{M}_S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

解　关于 $S \circ R$ 的矩阵是

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot M_S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$ ■

表示两个关系合成的矩阵可以用来求 $\mathbf{M}_{R^n}$ 的矩阵。特别地，由布尔幂的定义有

$$\mathbf{M}_{R^n} = \mathbf{M}_R^{[n]}$$

节末的练习 19 要求证明这个公式。

例 6　求关系 R^2 的矩阵表示，其中表示 R 的矩阵是

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

解　R^2 的矩阵是

$$\mathbf{M}_{R^2} = \mathbf{M}_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$ ■

6.3.3　用图表示关系

我们已经显示了一个关系可以通过列出它所有的有序对或使用 0-1 矩阵来表示。还有另一种重要的表示关系的方法就是图。把集合的每个元素表示成一个点，每个有序对表示成一条弧，弧上的箭头标明了弧的方向。当我们把一个有穷集上的关系看作一个有向图时，就可以使用这种图形表示。

定义 1 一个有向图由顶点（或结点）集 V 和边（或弧）集 E 组成，其中边集是 V 中元素的有序对的集合。顶点 a 叫做边(a,b)的始点，而顶点 b 叫做这条边的终点。

形如(a,a)的边用一条从顶点 a 到自身的弧表示。这种边叫做环。

例 7 具有顶点 a，b，c 和 d，边(a,b)，(a,d)，(b,b)，(b,d)，(c,a)，(c,d)和(d,b)的有向图给在图 6-5 中。■

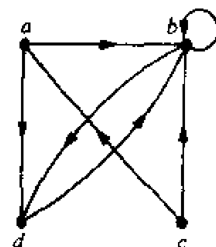


图 6-5 一个有向图

集合 A 上的关系 R 表示成一个有向图，这个图以 A 的元素作为顶点，以有序对(a,b)作为边，其中$(a,b)\in R$。这就在集合 A 上的关系和以 A 作为顶点集的有向图之间构成了一一对应。于是，每一个关于关系的论述就对应了一个关于有向图的论述，反之亦然。有向图给出了一个关于关系信息的可见显示。因此，也常常用图研究关系及其性质。（注意从集合 A 到集合 B 的关系可以用一个有向的图示来表示，其中对 A 中的每个元素存在一个顶点，B 中的每个元素存在一个顶点，如 6.1 节所示。然而，当 $A=B$ 时，那种图示表示与这里描述的有向图表示不同，它对关系信息的显示要少得多）。下面的例子说明怎样用有向图来表示关系。

例 8 集合 $\{1, 2, 3, 4\}$ 上的关系

$$R=\{(1,1),(1,3),(2,1),(2,3),(2,4),(3,1),(3,2),(4,1)\}$$

的有向图显示在图 6-6 中。■

例 9 图 6-7 中的有向图所表示的关系 R 中的有序对是什么？

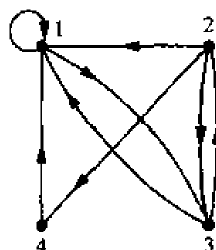


图 6-6 关系 R 的有向图

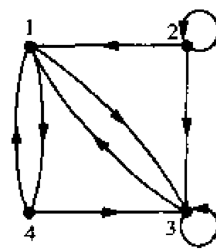


图 6-7 关系 R 的有向图

解 关系中的有序对(x,y)是

$$R=\{(1,3),(1,4),(2,1),(2,2),(2,3),(3,1),(3,3),(4,1),(4,3)\}$$

每个有序对对应有向图的一条边，$(2,2)$和$(3,3)$对应于环。■

表示关系的有向图可以用来确定关系是否有各种性质。例如，一个关系是自反的，当且仅当有向图的每个顶点都有环，从而每个形如(x,x)的有序对都出现在关系中。一个关系是对称的，当且仅当对有向图不同顶点之间的每一条边都存在一条方向相反的边，从而只要(x,y)在关系中就有(y,x)在关系中。类似地，一个关系是反对称的，当且仅当在不同的两个顶点之间不存在两条方向相反的边。最后，一个关系是传递的，当且仅当只要存在一条从顶点 x 到顶点 y 的边和一条从顶点 y 到顶点 z 的边，就有一条从顶点 x 到顶点 z 的边（完成一个三角形，其中每条边都是具有正确方向的有向边）。

例 10 对于图 6-8 所示的有向图的关系，确定它们是否为自反的、对称的、反对称的和

传递的。

解 因为 R 的有向图的每个顶点都有环，因此它是自反的。R 既不是对称的也不是反对称的，因为存在一条从 a 到 b 的边但没有从 b 到 a 的边，并且在连接 b 和 c 的两个方向都有边。最后，R 不是传递的，因为从 a 到 b 有边，从 b 到 c 有边，但是从 a 到 c 没有边。

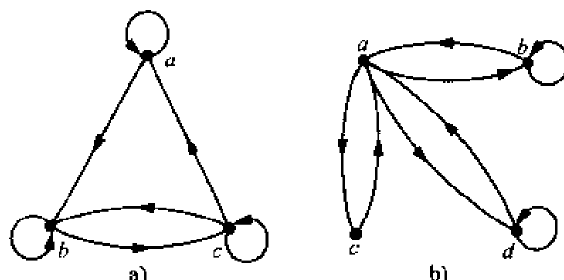


图 6-8 关系 R 和 S 的有向图

a) 关系 R 的有向图 b) 关系 S 的有向图

因为在有向图 S 的所有顶点没都出现环，因此关系不是自反的。因为在不同顶点之间的每条边都伴随一条方向相反的边，因此它是对称的但不是反对称的。从这个有向图也不难看出，S 不是传递的，因为 (c,a) 和 (a,b) 属于 S，但 (c,b) 不属于 S。 ■

练习

1. 用矩阵表示 $\{1,2,3\}$ 上的下面每个关系（按增序列出这个集合的元素）。
   a) $\{(1,1),(1,2),(1,3)\}$
   b) $\{(1,2),(2,1),(2,2),(3,3)\}$
   c) $\{(1,1),(1,2),(1,3),(2,2),(2,3),(3,3)\}$
   d) $\{(1,3),(3,1)\}$
2. 列出和下面矩阵对应的 $\{1,2,3\}$ 上的关系中的有序对（其中行和列对应于按增序列出的整数）：
   a) $\begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix}$ b) $\begin{bmatrix}0&1&0\\0&1&0\\0&1&0\end{bmatrix}$ c) $\begin{bmatrix}1&1&1\\1&0&1\\1&1&1\end{bmatrix}$
3. 怎样用关系的矩阵确定关系是否为反自反的？
4. 确定练习 2 中的矩阵所表示的关系是否为自反的、反自反的、对称的、反对称的和传递的。
5. 当 R 是有穷集 A 上的关系时，怎样从表示 R 的矩阵找到这个关系的补 $\overline{R}$ 的矩阵？
6. 当 R 是有穷集 A 上的关系时，怎样从表示 R 的矩阵找到这个关系的逆 R^{-1} 的矩阵？
7. 设 R 是矩阵

$$\mathbf{M}_R=\begin{bmatrix}0&1&1\\1&1&0\\1&0&1\end{bmatrix}$$

   所表示的关系。求表示下述关系的矩阵。

   a) R^{-1} b) $\overline{R}$ c) R^2

8. 设 R_1 和 R_2 是集合 A 上的关系并由矩阵

$$\mathbf{M}_{R_1}=\begin{bmatrix}0&1&0\\1&1&1\\1&0&0\end{bmatrix} \text{和 } \mathbf{M}_{R_2}=\begin{bmatrix}0&1&0\\0&1&1\\1&1&1\end{bmatrix}$$

表示。求表示下述关系的矩阵。

a) $R_1\cup R_2$ b) $R_1\cap R_2$ c) $R_2\circ R_1$ d) $R_1\circ R_1$ e) $R_1\oplus R_2$

9. 设 R 是矩阵

$$\mathbf{M}_R=\begin{bmatrix}0&1&0\\0&0&1\\1&1&0\end{bmatrix}$$

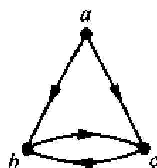
表示的关系。求表示下述关系的矩阵。

a) R^2 b) R^3 c) R^4

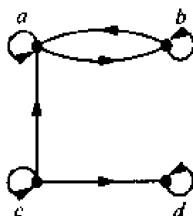
10. 对练习 1 的每个关系画出表示它们的有向图。
11. 对练习 2 的每个关系画出表示它们的有向图。
12. 画出表示关系$\{(a,a),(a,b),(b,c),(c,b),(c,d),(d,a),(d,b)\}$的有向图。

在练习 13～15 列出由有向图所表示的关系中的有序对。

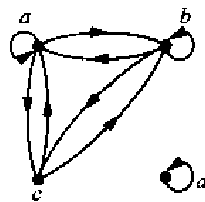
13.



14.



15.



16. 怎样用有穷集 A 上关系 R 的有向图确定关系是否是反自反的?
17. 对练习 13～15 所示的有向图表示的关系确定它们是否为自反的、反自反的、对称的、反对称的和传递的。
18. 给定两个表示关系的有向图，怎样找出这些关系的并、交、对称差、差以及合成的有向图?
19. 证明如果 $\mathbf{M}_R$ 是表示关系 R 的矩阵，那么 $\mathbf{M}_R^{[n]}$是表示关系 R^n 的矩阵。

6.4 关系的闭包

6.4.1 引言

一个计算机网络在波士顿、芝加哥、丹佛、底特律、纽约和圣地亚哥设有数据中心。从波士顿到芝加哥，波士顿到底特律，芝加哥到底特律，底特律到丹佛和纽约到圣地亚哥，都有单向的电话线。如果存在一条从数据中心 a 到 b 的电话线，(a,b)就属于关系 R。我们怎样确定从一个中心是否有一条电话线或多条电话线（可能不直接）链接到另一个中心？由于所有的链接不一定是直接的，例如从波士顿可通过底特律到丹佛，因此不能直接使用 R 来回答这个问题。用关系的语言说，R 不是传递的，因而它不包含可能被链接的所有的对。

正如我们将在这节证明的，可以通过构造包含 R 的最小的传递关系来找出每一对有着联线的数据中心。这个关系叫做 R 的传递闭包。

一般说来，设 R 是集合 A 上的关系。R 可能具有或者不具有某种性质 **P**，例如自反性、对称性或传递性。如果存在包含 R 的具有性质 **P** 的关系 S，并且 S 是包含 R 且具有性质 **P** 的每一个关系的子集，那么 S 叫做 R 的关于 **P** 的闭包。(注意一个关系关于一个性质的闭包可能不存在，见节末的练习 15 和 35。) 我们将说明怎样找关系的自反闭包、对称闭包和传递闭包。

6.4.2 闭包

集合 $A=\{1,2,3\}$ 上的关系 $R=\{(1,1),(1,2),(2,1),(3,2)\}$ 不是自反的。我们怎样可以产生一个包含 R 的尽可能小的自反关系呢？这可以通过把 $(2,2)$ 和 $(3,3)$ 加到 R 中来做到，因为只有它们是不在 R 中的形如 (a,a) 的对。很清楚，这个新关系包含了 R。此外，任何包含 R 的自反关系必须包含 $(2,2)$ 和 $(3,3)$。因为这个关系包含了 R，是自反的，并且被包含在每一个包含 R 的自反关系之中，因此它就是 R 的自反闭包。

正如这个例子所显示的，给定集合 A 上的关系 R，对于 $a\in R$，可以通过把形如 (a,a) 的所有的对，除了已在 R 中的之外，都加到 R 中，就构成了 R 的自反闭包。加入这些对产生了一个新的自反的、包含 R 的关系，并且它被包含在任何包含 R 的自反关系之中。我们看到 R 的自反闭包等于 $R\cup\Delta$，其中 $\Delta=\{(a,a)\mid a\in A\}$ 是 A 上的对角线关系。(读者应能验证这个结果。)

例 1 整数集上的关系 $R=\{(a,b)\mid a<b\}$ 的自反闭包是什么？

解 R 的自反闭包是

$$R\cup\Delta=\{(a,b)\mid a<b\}\cup\{(a,a)\mid a\in Z\}=\{(a,b)\mid a\leqslant b\}$$ ■

$\{1,2,3\}$ 上的关系 $\{(1,1),(1,2),(2,2),(2,3),(3,1),(3,2)\}$ 不是对称的。我们怎样可以产生一个包含 R 的尽可能小的对称关系？为此我们只需增加 $(2,1)$ 和 $(1,3)$，因为只有它们是具有 $(a,b)\in R$、而 (b,a) 不在 R 中的那种 (b,a) 对。这个新关系是对称的，且包含了 R。此外，任何包含了 R 的对称关系一定包含这个新关系，因为一个包含 R 的对称关系必须包含 $(2,1)$ 和 $(1,3)$。因此这个新关系叫做 R 的对称闭包。

正如这个例子所显示的，关系 R 的对称闭包可以通过增加所有形如 (b,a) 的有序对构成，其中 (a,b) 在关系中而 (b,a) 不在关系中。增加这些有序对产生一个关系，它是对称的，包含了 R，并且它被包含在任何包含 R 的对称关系之中。关系 R 的对称闭包可以通过取关系与它的逆的并来构造，即 $R\cup R^{-1}$ 是 R 的对称闭包，其中 $R^{-1}=\{(b,a)\mid(a,b)\in R\}$。读者应该验证这个结果。

例 2 正整数集合上的关系 $\{(a,b)\mid a>b\}$ 的对称闭包是什么？

解 R 的对称闭包是关系

$$R\cup R^{-1}=\{(a,b)\mid a>b\}\cup\{(b,a)\mid a>b\}=\{(a,b)\mid a\neq b\}$$ ■

假设关系 R 不是传递的，我们怎样产生一个包含 R 的传递关系并使得这个新的关系被

包含在任何包含 R 的传递关系之中？对于已经在 R 中的 (a,b) 和 (b,c)，可以通过增加所有形如 (a,c) 的有序对构成 R 的传递闭包吗？考虑集合 $\{1,2,3,4\}$ 上的关系 $R=\{(1,3),(1,4),(2,1),(3,2)\}$，这个关系不是传递的，因为对于在 R 中的 (a,b) 和 (b,c)，它不包含所有形如 (a,c) 的有序对。这种不在 R 中的有序对是 $(1,2)$，$(2,3)$，$(2,4)$ 和 $(3,1)$。把这些有序对加到 R 中并不能产生一个传递关系，因为所得的结果关系包含 $(3,1)$ 和 $(1,4)$，但不包含 $(3,4)$。这说明构造关系的传递闭包比起构造它们的自反闭包或对称闭包更复杂。下面将建立一个构造传递闭包的算法。正如将要显示的，一个关系的传递闭包可以通过增加那些必须出现的有序对来得到，要重复这个过程直到没有必须增加的有序对为止。

6.4.3 有向图的路径

我们将看到用有向图表示关系有助于构造关系的传递闭包。为此现在引入某些将要用到的术语。

通过沿有向图的边（按照这条边的箭头指示的相同方向）旅行就得到一条有向图中路径。

定义 1 在有向图 G 中，从 a 到 b 的一条路径是 G 中一条或多条边的序列 (x_0,x_1)，(x_1,x_2)，(x_2,x_3)，…，(x_{n-1},x_n)，其中 $x_0=a$、$x_n=b$，即一个边的序列，其中路径中一条边的终点和下一条边的始点相同。这条路径记为 x_0，x_1，…，x_{n-1}，x_n，长度为 n。在同一顶点开始和结束的**路径**叫做**回路**或**圈**。

有向图的一条路径可以多次通过一个顶点。此外，有向图的一条边也可以多次出现在一条路径中。读者应该注意到某些作者允许长度为 0 的路径，即不含边的路径。这本书中所有的路径的长度必须至少是 1。

例 3 下面哪些路径是图 6-9 的有向图中的路径：a，b，e，d；a，e，c，d，b；b，a，c，b，a，a，b；d，c；c，b，a；e，b，a，b，a，b，e？这些路径的长度是多少？上述路径中哪些路径是回路？

解 因为 (a,b)、(b,e) 和 (e,d) 都是边，a、b、e、d 是长为 3 的路径。因为 (c,b) 不是边，a、e、c、d、b 不是路径。因为 (b,a)、(a,c)、(c,b)、(b,a)、(a,a) 和 (a,b) 都是边，b、a、c、b、a、a、b 是长为 6 的路径。我们也看到，因为 (d,c) 是边，d、c 是长为 1 的路径。还有由于 (c,d) 和 (b,a) 是边，c、b、a 是长为 2 的路径。(e,b)、(b,a)、(a,b)、(b,a)、(a,b)、(b,e) 都是边，因此 e、b、a、b、a、b、e 是长为 6 的路径。

两条路径 b、a、c、b、a、a、b 和 e、b、a、b、a、b、e 是回路，因为它们在同一顶点开始和结束。路径 a、b、e、d、c、b、a 和 d、c 不是回路。■

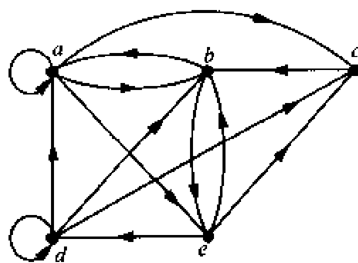


图 6-9 一个有向图

术语路径也用于关系。把有向图的定义推广到关系可知，如果存在一个元素的序列 a，x_1，x_2，…，x_{n-1}，b 使得 $(a,x_1)\in R$，$(x_1,x_2)\in R$，…，$(x_{n-1},b)\in R$，那么在 R 中存在一条从 a 到 b 的路径。从关系中的路径定义可以得到下面的定理。

定理 1 设 R 是集合 A 上的关系。从 a 到 b 存在一条长为 n 的路径，当且仅当 $(a,b)\in R^n$。

证 我们将使用数学归纳法。根据定义，从 a 到 b 存在一条长为 1 的路径，当且仅当 $(a,b)\in R$。因此当 $n=1$ 时定理为真。

假定对于正整数 n 定理为真，这是归纳假设。从 a 到 b 存在一条长为 $n+1$ 的路径，当且仅当存在元素 $c\in A$，使得从 a 到 c 存在一条长为 1 的路径，即 $(a,c)\in R$，以及一条从 c 到 b 的长为 n 的路径，即 $(c,b)\in R^n$。因此，由归纳假设，从 a 到 b 存在一条长为 $n+1$ 的路径，当且仅当存在一个元素 c，使得 $(a,c)\in R$ 和 $(c,b)\in R^n$。但是存在这样一个元素，当且仅当 $(a,b)\in R^{n+1}$。因此，从 a 到 b 存在一条长为 $n+1$ 的路径，当且仅当 $(a,b)\in R^{n+1}$，定理得证。 □

6.4.4 传递闭包

现在证明找一个关系的传递闭包与在相关的有向图中确定哪些顶点对被路径联结是等价的。注意到这一点，我们定义一个新的关系。

定义 2 设 R 是集合 A 上的关系。连通关系 R^* 由对 (a,b) 构成，使得在 R 中从顶点 a 到 b 之间存在一条路径。

因为 R^n 由对 (a,b) 构成，使得存在一条从 a 到 b 的长为 n 的路径，从而 R^* 是所有集合 R^n 的并。换句话说，

$$R^*=\bigcup_{n=1}^{\infty}R^n$$

许多模型都用到关系的连通性。

例 4 设 R 是世界上所有人的集合上的关系，如果 a 认识 b，那么 R 包含 (a,b)。R^n 是什么？其中 n 是大于 2 的正整数。R^* 是什么？

解 如果存在人 c，使得 $(a,c)\in R$ 且 $(c,b)\in R$，即存在人 c 使得 a 认识 c，c 认识 b，那么关系 R^2 包括 (a,b)。类似地，如果存在人 x_1，x_2，…，x_{n-1} 使得 a 认识 x_1，x_1 认识 x_2，…，x_{n-1} 认识 b，那么 R 包含对 (a,b)。

如果存在人的序列，从 a 开始至 b 终止，使得序列中的每个人都认识序列中的下一个人，那么 R^* 包含对 (a,b)。（关于 R^* 存在许多有趣的猜想。你认为这个连通关系包含以你作为第一元素、蒙古的总统作为第二元素的一对元素吗?） ■

例 5 设 R 是纽约市所有地铁站的集合上的关系。如果可以从站 a 不换车就旅行到站 b，那么 R 包含对 (a,b)。当 n 是正整数时，R^n 是什么？R^* 是什么?

解 如果经过至多 $n-1$ 次换车就可以从站 a 旅行到站 b，关系 R^n 就包含 (a,b)。从站 a 旅行到站 b，如果需要可以换车任意多次，关系 R^* 就由这种有序对 (a,b) 组成。（读者应该能够验证这些论断。） ■

例 6 设 R 是美国所有的州的集合上的关系。如果州 a 和州 b 有公共的边界，那么 R

包含(a,b)。R^n 是什么？其中 n 是正整数。R^* 是什么？

解 关系 R^n 由对(a,b)构成，其中可以从州 a 恰好跨越 n 个边界到州 b。R^* 由对(a,b)构成，其中可以从州 a 跨越任意多次的边界到州 b。(读者应该能够验证这些论断。) 只有包含不连接到美国大陆的州（即含有阿拉斯加或夏威夷）的有序对是不在 R^* 中的对。■

下面的定理证明一个关系的传递闭包和相关的连通性关系是相同的。

定理 2 关系 R 的传递闭包等于连通关系 R^*。

证 注意 R^* 包含 R。为证明 R^* 是 R 的传递闭包必须证明 R^* 是传递的，且对一切包含 R 的传递关系 S 有 $R^* \subseteq S$。

首先，我们证明 R^* 是传递的。如果$(a,b)\in R^*$和$(b,c)\in R^*$，那么在 R 中存在从 a 到 b 和从 b 到 c 的路径。我们以从 a 到 b 的路径开始，并且接着沿从 b 到 c 的路径就得到一条从 a 到 c 的路径。因此，$(a,c)\in R^*$。这就得出 R^* 是传递的。

现在假设 S 是包含 R 的传递关系。因为 S 是传递的，S^n 也是传递的（读者应该能验证这一点），并且 $S^n \subseteq S$（由 6.1 节定理 1）。此外，因为

$$S^* = \bigcup_{k=1}^{\infty} S^k$$

和 $S^k \subseteq S$，因此 $S^* \subseteq S$。现在注意到如果 $R \subseteq S$，那么 $R^* \subseteq S^*$，这是由于任何 R 中的路径也是 S 中的路径。从而 $R^* \subseteq S^* \subseteq S$。于是，任何包含 R 的传递关系也一定包含 R^*。因此，R^* 是 R 的传递闭包。□

现在知道传递闭包等于连通关系，我们考虑这个关系的计算问题。在一个有限的有向图中不需要检测任意长的路径来确定是否在两个顶点之间存在一条路径。正如下面的引理所证明的，检测包含不超过 n 条边的路径就足够了，这里 n 是集合中的元素个数。

引理 1 设 A 是 n 元素集合，R 是 A 上的关系。如果 R 中存在一条从 a 到 b 的路径，那么存在一条长度不超过 n 的这种路径。此外，当 $a \neq b$ 时，如果在 R 中存在一条从 a 到 b 的路径，那么存在一条长度不超过 $n-1$ 的这种路径。

证 假设 R 中存在一条从 a 到 b 的路径。令 m 是这种路径的最短长度。假设 x_0，x_1，x_2，…，x_{m-1}，x_m 是一条这样的路径，其中 $x_0 = a$，$x_m = b$。

假设 $a = b$ 和 $m > n$，使得 $m \geq n+1$。由鸽巢原理，因为 A 中有 n 个顶点，在 m 个顶点 x_0，x_1，x_2，…，x_{m-1}中至少有两个是相同的（见图 6-10）。

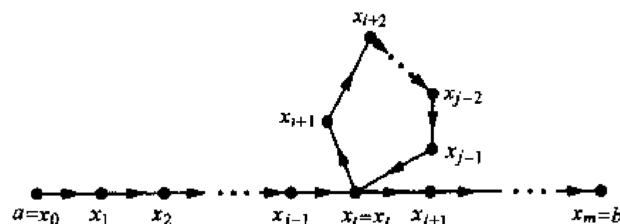


图 6-10 产生一条长度不超过 n 的路径

假设 $x_i=x_j$，$0\leqslant i<j\leqslant m-1$。那么这条路径包含一条从 x_i 到 x_i 自身的回路。可以把这条回路从由 a 到 b 的路径中删除，剩下的路径，即 x_0，x_1，…，x_i，x_{j+1}，…，x_{m-1}，x_m，是从 a 到 b 的更短的路径。因此，最短长度的这种路径的长度一定小于或等于 n。

$a\neq b$ 的情况留给读者作为练习。□

由引理 1，我们看出 R 的传递闭包是 R，R^2，R^3，…，R^n 的并。这是由于在 R^* 的两个顶点之间存在一条路径，当且仅当对某个正整数 i（$i\leqslant n$）在 R^i 的这些顶点之间存在一条路径。因为

$$R^*=R\cup R^2\cup R^3\cup\cdots\cup R^n$$

并且表示关系的并的 0-1 矩阵是这些关系的 0-1 矩阵的联合，因此对于传递闭包的 0-1 矩阵是 R 的 0-1 矩阵的前 n 次幂的 0-1 矩阵的联合。

定理 3　设 $\mathbf{M}_R$ 是 n 元素集合上的关系 R 的 0-1 矩阵。那么传递闭包 R^* 的 0-1 矩阵是

$$\mathbf{M}_{R^*}=\mathbf{M}_R\vee\mathbf{M}_R^{[2]}\vee\mathbf{M}_R^{[3]}\vee\cdots\vee\mathbf{M}_R^{[n]}$$

例 7　求关系 R 的传递闭包的 0-1 矩阵，其中

$$\mathbf{M}_R=\begin{bmatrix}1&0&1\\0&1&0\\1&1&0\end{bmatrix}$$

解　由定理 3，R^* 的 0-1 矩阵是

$$\mathbf{M}_{R^*}=\mathbf{M}_R\vee\mathbf{M}_R^{[2]}\vee\mathbf{M}_R^{[3]}$$

因为

$$\mathbf{M}_R^{[2]}=\begin{bmatrix}1&1&1\\0&1&0\\1&1&1\end{bmatrix}\quad 和\quad \mathbf{M}_R^{[3]}=\begin{bmatrix}1&1&1\\0&1&0\\1&1&1\end{bmatrix}$$

所以

$$\mathbf{M}_{R^*}=\begin{bmatrix}1&0&1\\0&1&0\\1&1&0\end{bmatrix}\vee\begin{bmatrix}1&1&1\\0&1&0\\1&1&1\end{bmatrix}\vee\begin{bmatrix}1&1&1\\0&1&0\\1&1&1\end{bmatrix}=\begin{bmatrix}1&1&1\\0&1&0\\1&1&1\end{bmatrix}$$ ■

定理 3 可以作为计算关系 R^* 的矩阵的算法基础。为求出这个矩阵，要连续计算 $\mathbf{M}_R$ 的布尔幂，直到第 n 次幂为止。当计算每次幂时就构成这个幂与所有较小的幂的联合。当做到第 n 次幂时，就得到关于 R^*的矩阵。这个过程显示在算法 1。

我们可以容易地求出用算法 1 确定关系的传递闭包所使用的位运算次数。计算布尔幂 $\mathbf{M}_R$，$\mathbf{M}_R^{[2]}$，…，$\mathbf{M}_R^{[n]}$需要找到 $n-1$ 个 $n\times n$ 的 0-1 矩阵的布尔积。每个布尔积可以使用 $n^2(2n-1)$次位运算求得。因此，计算这些乘积使用$n^2(n-1)(n-1)$次位运算。

为从 n 个 $\mathbf{M}_R$ 的布尔幂求 $\mathbf{M}_{R^*}$，需要求 $n-1$ 个 0-1 矩阵的联合。计算每一个联合使用 n^2 次位运算。因此，在这部分计算中使用 $(n-1)n^2$ 次位运算。所以，当使用算法 1 时，可以用 $n^2(2n-1)(n-1)+(n-1)n^2=2n^3(n-1)=O(n^4)$次位运算就可以求出 n

元素集合上关系的传递闭包的矩阵。下面我们将要描述一个更有效的求传递闭包的算法。

算法 1　计算传递闭包的过程

```
procedure transitive closure (M_R: n×n 的 0-1 矩阵)
A := M_R
B := A
for i := 2 to n
begin
    A := A⊙M_R
    B := B∨A
end {B 是关于 R* 的 0-1 矩阵}
```

6.4.5　沃舍尔算法

沃舍尔算法，是由于史蒂芬·沃舍尔[⊖]面命名的，是他在 1960 年给出的算法。这个算法是计算关系的传递闭包的有效方法。算法 1 求出 n 元素集合上关系的传递闭包使用 $2n^3(n-1)$ 次位运算，而沃舍尔算法只使用 $2n^3$ 次位运算就可以求出。

注意　沃舍尔算法有时也叫做罗伊-沃舍尔算法，因为罗伊（B. Roy）在 1959 年描述了这个算法。

假设 R 是 n 元集上的关系，设 $v_1, v_2, \cdots, v_n$ 是这 n 个元素的任意排列。沃舍尔算法中用到一条路径的内点的概念。如果 $a, x_1, x_2, \cdots, x_{m-1}, b$ 是一条路径，它的内点是 $x_1, x_2, \cdots, x_{m-1}$，即除了第一和最后一个顶点之外出现在路径中的所有顶点。例如，在有向图的一条路径 a, c, d, f, g, h, b, j 的内点是 c, d, f, g, h, b。a, c, d, a, f, b 的内点是 c, d, a, f。（注意这条路径的始点不是内点，除非这条路径再次访问它，且不是作为终点来访问的。类似地，这条路径的终点也不是内点，除非它在这之前曾被这条路径访问过，且不是作为始点来访问的。）

⊖ 史蒂芬·沃舍尔(Stephen Warshall，生于 1935 年)　沃舍尔生于纽约，在布鲁克莱恩的公立学校接受教育。他进入哈佛大学学习，1956 年获得数学学位。他没有得到更高的学位，因为在那个年代在他感兴趣的领域没有合适的培养计划。但是，他在几个不同的大学修了研究生课程并且对计算机科学和软件工程的发展作出了贡献。从哈佛毕业以后，沃舍尔在 ORO（运筹学办公室）工作，它是由约翰·霍普金斯建立的并为美国陆军作研究和开发工作。1958 年他离开 ORO 去一个叫做技术操作的公司工作，在那里他帮助建立了一个从事军事软件课题研究和开发的实验室。1961 年他离开了技术操作公司去创建马萨诸塞计算机联合公司。后来，这个公司变成了应用数据研究（ADR）公司的一部分。在这次合并以后，沃舍尔进入了 ADR 的董事会并且管理了各种项目和组织。他在 1982 年从 ADR 退休。

在任职期间，沃舍尔在操作系统、编译设计、语言设计和运筹学领域从事研究和开发工作。在 1971～1972 年学术年会上他在法国各大学作了关于软件工程方面的报告。关于这个传递闭包算法，即目前所称的沃舍尔算法的正确性证明有一个有趣的故事。他和一个在技术操作公司的同事打赌，谁首先确定这个算法是否永远有效谁就赢得一瓶甜酒。沃舍尔过了一夜给出了他的证明，赢得了这瓶甜酒，并和这个打赌的同事共同分享了这瓶甜酒。因为沃舍尔不喜欢坐在写字桌旁，他的许多创造性工作都是在不寻常的地方完成的，例如，在印度洋的一艘帆船上或在希腊的一棵柠檬树下。

沃舍尔算法的基础是构造一系列 0－1 矩阵。这些矩阵是 $\mathbf{W}_0$，$\mathbf{W}_1$，… $\mathbf{W}_n$，其中 $\mathbf{W}_0 = \mathbf{M}_R$ 是这个关系的 0－1 矩阵，且 $\mathbf{W}_k = [w_{ij}^{[k]}]$。如果存在一条从 v_i 到 v_j 的路径使得这条路径的所有内点都在集合 $\{v_1, v_2 \cdots, v_k\}$（表中的前 k 个顶点）之中，那么 $w_{ij}^{[k]} = 1$，否则为 0。（这条路径的始点和终点可能在表中的前 k 个顶点的集合之外。）注意 $\mathbf{W}_n = \mathbf{M}_{R^*}$，因为 $\mathbf{M}_{R^*}$ 的第 (i, j) 项是 1，当且仅当存在一条从 v_i 到 v_j 的路径，且全部内点都在集合 $\{v_1, v_2, \cdots, v_n\}$ 之中（但这些就是有向图的所有顶点）。下面的例子说明矩阵 $\mathbf{W}_k$ 表示什么。

例 8　设 R 是一个关系，它的有向图如图 6-11 所示。设 a，b，c，d 是集合元素的排列。求矩阵 $\mathbf{W}_0$，$\mathbf{W}_1$，$\mathbf{W}_2$，$\mathbf{W}_3$ 和 $\mathbf{W}_4$。矩阵 $\mathbf{W}_4$ 是 R 的传递闭包。

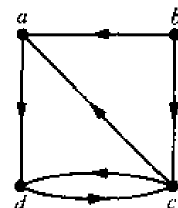


图 6-11　关系 R 的有向图

解　令 $v_1 = a$，$v_2 = b$，$v_3 = c$，且 $v_4 = d$。$\mathbf{W}_0$ 是这个关系的矩阵，于是

$$\mathbf{W}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

如果存在一条 v_i 到 v_j 的且只有 $v_1 = a$ 作为内点的路径，$\mathbf{W}_1$ 的 (i, j) 项有 1。注意因为所有长为 1 的路径没有内点，所以仍旧可以使用这些路径。此外存在一条从 b 到 d 的路径，即 b，a，d。因此

$$\mathbf{W}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

如果存在一条从 v_i 到 v_j 的且只有 $v_1 = a$ 和 $v_2 = b$ 作为内点的路径，$\mathbf{W}_2$ 的 (i, j) 项有 1。因为没有边以 b 作为终点，当允许 b 作为内点时不会得到新的路径。因此，$\mathbf{W}_2 = \mathbf{W}_1$。

若存在一条从 v_i 到 v_j 的只用 $v_1 = a$，$v_2 = b$ 和 $v_3 = c$ 作为内点的路径，则 $\mathbf{W}_3$ 的 (i, j) 项有 1。我们现在有从 d 到 a 的路径，即 d，c，a 和从 d 到 d 的路径 d，c，d。因此

$$\mathbf{W}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

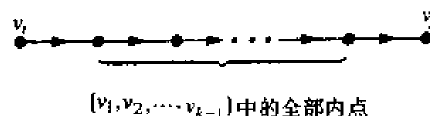
最后，如果存在一条从 v_i 到 v_j 的路径，并且以 $v_1 = a$，$v_2 = b$，$v_3 = c$ 及 $v_4 = d$ 作为内点，那么 $\mathbf{W}_4$ 的 (i, j) 项为 1。因为这些是图的全部顶点，此项为 1，当且仅当存在一条从 v_i 到 v_j 的路径。因此

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

这个最后的矩阵就是传递闭包的矩阵。 

沃舍尔算法通过有效的计算 $\mathbf{W}_0=\mathbf{M}_R$，$\mathbf{W}_1$，$\mathbf{W}_2$，…，$\mathbf{W}_n=\mathbf{M}_{R^*}$ 来计算 $\mathbf{M}_{R^*}$。不难看出可以直接从 $\mathbf{W}_{k-1}$ 计算 $\mathbf{W}_k$：存在一条从 v_i 到 v_j 的只以 v_1，v_2，…，v_k 中的顶点作为内点的路径，当且仅当或者存在一条从 v_i 到 v_j 的且内点是表中前 $k-1$ 个顶点的路径，或者存在从 v_i 到 v_k 的路径和从 v_k 到 v_j 的路径，而这些路径的内点是表中的前 $k-1$ 个顶点。这就是说，或者在 v_k 被允许作为内点之前从 v_i 到 v_j 已经存在一条路径，或者允许 v_k 作为内点产生一条从 v_i 到 v_k 然后从 v_k 到 v_j 的路径。这两种情况显示在图 6-12 中。

情况1



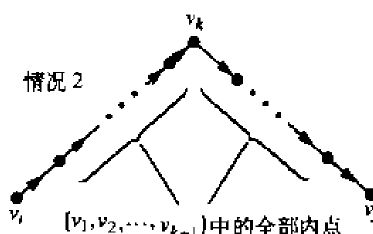


图 6-12　把 v_k 加到允许使用的内点集中

第一种类型的路径存在，当且仅当 $w_{ij}^{[k-1]}=1$，且第二种类型的路径存在，当且仅当 $w_{ik}^{[k-1]}=1$ 和 $w_{kj}^{[k-1]}=1$。于是，$w_{ij}^{[k]}$ 是 1，当且仅当或者 $w_{ij}^{[k-1]}=1$，或者 $w_{ik}^{[k-1]}=1$ 和 $w_{kj}^{[k-1]}=1$。这就得到下面的引理。

引理 2　设 $\mathbf{W}_k=[w_{ij}^{[k]}]$ 是 0－1 矩阵，它在 (i,j) 位置有 1，当且仅当存在一条从 v_i 到 v_j 的内点取自集合 $\{v_1,v_2,\cdots,v_k\}$ 的路径。那么

$$w_{ij}^{[k]}=w_{ij}^{[k-1]}\vee(w_{ik}^{[k-1]}\wedge w_{kj}^{[k-1]})$$

其中 i，j 和 k 是不超过 n 的正整数。

引理 2 提供了有效计算矩阵 $\mathbf{W}_k$（$k=1$，2，…，n）的手段。我们使用引理 2 把沃舍尔算法的伪码给在算法 2 中。

算法 2　沃舍尔算法

```
procedure Warshall (M_R: n×n 的 0-1 矩阵)
W := M_R
For k := 1 to n
begin
    for i := 1 to n
    begin
        for j := 1 to n
            w_ij = w_ij ∨ (w_ik ∧ w_kj)
    end
end {W = [w_ij] 是 M_R*}
```

很容易以位运算次数计算出沃舍尔算法的计算复杂性。使用引理 2，从项 $w_{ij}^{[k-1]}$、$w_{ik}^{[k-1]}$ 和 $w_{kj}^{[k-1]}$ 找到项 $w_{ij}^{[k]}$ 需要 2 次位运算。为从 $\mathbf{W}_{k-1}$ 求出 $\mathbf{W}_k$ 的所有的 n^2 个项需要 $2n^2$ 次位运算。因为沃舍尔算法从 $\mathbf{W}_0=\mathbf{M}_R$ 开始，计算 n 个 0-1 矩阵的序列 $\mathbf{W}_1$，$\mathbf{W}_2$，…，

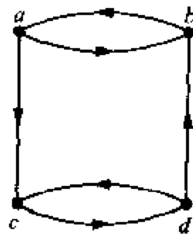
$W_n = M_R^*$，使用的位运算总数是 $n \cdot 2n^2 = 2n^3$。

练习

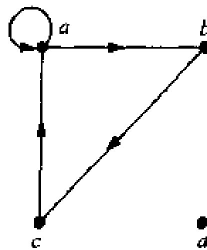
1. 设 R 是集合{0,1,2,3}上的关系，R 包含有序对(0,1),(1,1),(1,2),(2,0),(2,2)和(3,0)。求
   a) R 的自反闭包。　　b) R 的对称闭包。
2. 设 R 是整数集上的关系{$(a,b)\mid a\neq b$}。R 的自反闭包是什么？
3. 设 R 是整数集上的关系{$(a,b)\mid a$ 整除 b}。R 的对称闭包是什么？
4. 从有穷集上关系的有向图怎样构造表示它的自反闭包的有向图？

在练习 5～7 中画出给定有向图所表示关系的自反闭包的有向图。

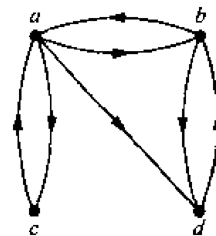
5.



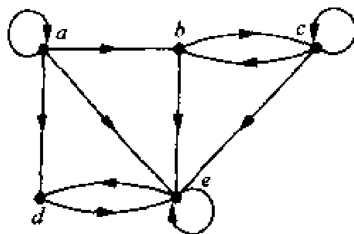
6.



7.



8. 从有穷集上关系的有向图怎样构造表示它的对称闭包的有向图？
9. 对于练习 5～7 的有向图表示的关系，找出关系的对称闭包的有向图。
10. 找出包含了例 2 中关系的最小的自反和对称的关系。
11. 对于练习 5～7 的每个有向图表示的关系，求包含它的最小的自反对称关系的有向图。
12. 假设有穷集 A 上的关系 R 由矩阵 $\mathbf{M}_R$ 表示，证明表示 R 的自反闭包的矩阵是 $\mathbf{M}_R \vee \mathbf{I}_n$。
13. 假设有穷集上 A 的关系 R 由矩阵 $\mathbf{M}_R$ 表示，证明表示 R 的对称闭包的矩阵是 $\mathbf{M}_R \vee \mathbf{M}_R^t$。
14. 证明关系 R 关于性质 **P** 的闭包如果存在，就是所有包含 R 的具有性质 **P** 的关系的交。
15. 什么时候可能定义一个关系的反自反闭包，即包含 R 的一个反自反关系且被包含在每一个包含 R 的反自反关系之中？
16. 确定下面的顶序列是否为下面的有向图中的路径。



   a) a, b, c, e
   b) b, e, c, b, e
   c) a, a, b, e, d, e
   d) b, c, e, d, a, a, b
   e) b, c, c, b, e, d, e, d
   f) $a, a, b, b, c, c, b, e, d$
17. 求出练习 16 的有向图中所有长为 3 的路径。
18. 确定练习 16 的有向图中是否存在一条以下面给定的第一顶点作为始点和第二顶点作为终点的路径。
   a) a, b　　b) b, a　　c) b, b　　d) a, e　　e) b, d
   f) c, d　　g) d, d　　h) e, a　　i) e, c

19. 设 R 是集合{1,2,3,4,5}上的关系，R 包含有序对(1,3),(2,4),(3,1),(3,5),(4,3),(5,1),(5,2)和(5,4)。求

    a) R^2 b) R^3

    c) R^4 d) R^5

    e) R^6 f) R^*

20. 设 R 是关系，如果存在一条从 a 城到 b 城的直达航班，则 R 包含有序对(a,b)。什么时候(a,b)在下面的关系中?

    a) R^2 b) R^3 c) R^*

21. 设 R 是所有学生的集合上的关系，如果 $a \neq b$ 且 a 和 b 至少有一门公共课程，则 R 包含了有序对(a,b)。什么时候(a,b)在下面的关系中?

    a) R^2 b) R^3 c) R^*

22. 假设关系 R 是自反的。证明 R^* 是自反的。
23. 假设关系 R 是对称的。证明 R^* 是对称的。
24. 假设关系 R 是反自反的。关系 R^2 一定是反自反的吗?
25. 使用算法 1 找出下面{1,2,3,4}上的关系的传递闭包。

    a) {(1,2),(2,1),(2,3),(3,4),(4,1)}

    b) {(2,1),(2,3),(3,1),(3,4),(4,1),(4,3)}

    c) {(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)}

    d) {(1,1),(1,4),(2,1),(2,3),(3,1),(3,2),(3,4),(4,2)}

26. 使用算法 1 找出下面$\{a,b,c,d,e\}$上的关系的传递闭包。

    a) $\{(a,c),(b,d),(c,a),(d,b),(e,d)\}$

    b) $\{(b,c),(b,e),(c,e),(d,a),(e,b),(e,c)\}$

    c) $\{(a,b),(a,c),(a,e),(b,a),(b,c),(c,a),(c,b),(d,a),(e,d)\}$

    d) $\{(a,e),(b,a),(b,d),(c,d),(d,a),(d,c),(e,a),(e,b),(e,c),(e,e)\}$

27. 使用沃舍尔算法找出练习 25 中关系的传递闭包。
28. 使用沃舍尔算法找出练习 26 中关系的传递闭包。
29. 求出包含关系{(1,2),(1,4),(3,3),(4,1)}的最小的关系使得它是

    a) 自反的和传递的。

    b) 对称的和传递的。

    c) 自反的、对称的和传递的

30. 完成在引理 1 当 $a \neq b$ 的情况下的证明。
31. 已经设计出算法用 $O(n^{2.8})$次位运算来计算两个 $n \times n$ 的 0-1 矩阵的布尔积。假设可以使用这些算法，给出用算法 1 和沃舍尔算法求 n 元素集合上关系的传递闭包所用位运算次数的大 O 估计。

*32. 如果有向图的两个顶点间的最短路径存在，设计一个算法使用路径中的内点的概念求这种最短路径的长度。

33. 修改算法 1 找出 n 元素集合上关系的传递闭包的自反闭包。
34. 修改沃舍尔算法找出 n 元素集合上关系的传递闭包的自反闭包。
35. 证明集合{0,1,2}上的关系 R = {(0,0),(0,1),(1,1),(2,2)}关于下述性质 **P** 的闭包不存

在，如果 P 是

a）“不是自反的”。

b）“有奇数个元素”。

6.5　等价关系

6.5.1　引言

在学期开始之前学院的学生登记注册。姓的第一个字母从 A 到 G、从 H 到 N 和从 O 到 Z 的学生分别可以在上午 8 点到 11 点、上午 11 点到下午 2 点和下午 2 点到 5 点之间的任何时间注册。设 R 是关系，R 包含 (x, y)，当且仅当 x 和 y 是姓的第一个字母在同一组的学生。因此 x 和 y 可以在同一时间注册，当且仅当 (x, y) 属于 R。容易看出 R 是自反的、对称的和传递的。此外，依赖于他们的姓的第一个字母，R 把学生的集合划分成 3 类。为了解一个学生在什么时间注册，我们仅关心这个学生是在 3 个类中的哪个类，并不需要识别这个学生。

整数 a 和 b 有模 4 同余的关系，如果 4 整除 $a-b$。后面我们将证明这个关心是自反的、对称的和传递的。不难看出 a 和 b 有关系，当且仅当被 4 整除时 a 和 b 有相同的余数。这个关系将整数集划分成不同的 4 类。当我们仅关心一个整数被 4 整除时的余数时只需要知道它在哪个类而不必知道它的特定值。

这两个关系，R 和模 4 同余关系是等价关系，即自反、对称和传递关系的例子。在这一节将证明这种关系把集合划分成由等价元素构成的不相交的类。当我们仅关心集合的一个元素是否在某个元素类，而不介意它的特殊身份时，就出现了等价关系。

6.5.2　等价关系

在这一节我们将研究具有一组特殊性质的关系，可以用这组性质在某一方而类似的相关个体之间建立联系。

定义 1　集合 A 上的关系叫做等价的，如果它是自反的、对称的和传递的。

两个由等价关系联系起来的元素叫做等价的元素（因为等价关系是对称的，这个定义是有意义的。）由于等价关系是自反的，在一个等价关系中每个元素都与自身等价。此外，因为等价关系是传递的，如果 a 和 b 是等价的且 b 和 c 是等价的，那么 a 和 c 是等价的。

下面的例子说明了等价关系的概念。

例 1　设 R 是英语字母串的集合上的关系，并且 aRb，当且仅当 $l(a)=l(b)$，其中 $l(x)$ 是串 x 的长度。R 是等价关系吗？

解　因为 $l(a)=l(a)$，从而只要 a 是一个串，就有 aRb，故 R 是自反的。其次，假设 aRb，即 $l(a)=l(b)$，那么有 bRa，因为 $l(b)=l(a)$。因此 R 是对称的。最后，假设 aRb 和 bRc，那么有 $l(a)=l(b)$ 和 $l(b)=l(c)$。因此 $l(a)=l(c)$，即 aRc。从而 R 是传递的。由于 R 是自反的、对称的和传递的，R 是等价关系。■

例 2　设 R 是整数集上的关系，并且 aRb，当且仅当 $a=b$ 或 $a=-b$。在 6.1 节我们证

明了 R 是自反的、对称的和传递的。因此 R 是等价关系。■

例 3 设 R 是实数集上的关系，并且 aRb，当且仅当 $a-b$ 是整数。R 是等价关系吗？

解 因为对所有的实数 a，$a-a=0$ 是整数，即对所有的实数有 aRa，因此 R 是自反的。现在假设 aRb，那么 $a-b$ 是整数，所以 $b-a$ 也是整数。因此有 bRa。R 是对称的。如果 aRb 和 bRc，那么 $a-b$ 和 $b-c$ 是整数，所以 $a-c=(a-b)+(b-c)$ 也是整数。因此 aRc。于是，R 是传递的。综合上述，R 是等价关系。■

最广泛使用的等价关系之一是模 m 同余关系，其中 m 是大于 1 的正整数。

例 4 模 m 同余。设 m 是大于 1 的正整数。证明关系

$$R=\{(a,b)\mid a\equiv b\ (\mathrm{mod}\ m)\}$$

是整数集上的等价关系。

解 回顾2.3节，$a\equiv b\ (\mathrm{mod}\ m)$，当且仅当 m 整除 $a-b$。注意 $a-a=0$ 被 m 整除，因为 $0=0\cdot m$。因此 $a\equiv a\ (\mathrm{mod}\ m)$，从而模 m 同余关系是自反的。现在假设 $a\equiv b\ (\mathrm{mod}\ m)$，那么 $a-b$ 被 m 整除，即 $a-b=km$，其中 k 是整数。从而 $b-a=(-k)m$，即 $b\equiv a\ (\mathrm{mod}\ m)$。因此模 m 同余关系是对称的。下面假设 $a\equiv b\ (\mathrm{mod}\ m)$ 和 $b\equiv c\ (\mathrm{mod}\ m)$，那么 m 整除 $a-b$ 和 $b-c$。因此，存在整数 k 和 l 使得 $a-b=km$ 和 $b-c=lm$。把这两个等式加起来得 $a-c=(a-b)+(b-c)=km+lm=(k+1)m$。于是，$a\equiv c\ (\mathrm{mod}\ m)$。从而，模 m 同余关系是传递的。综合上述，模 m 同余关系是等价关系。■

6.5.3 等价类

设 A 是在你们学校所有的高中毕业生。考虑 A 上的关系 R，R 由所有的对 (x,y) 构成，其中 x 和 y 从同一高中毕业。给定学生 x，我们可以构造所有的关于 R 与 x 等价的学生的集合。这个集合由与 x 在同一高中毕业的所有学生构成。A 的这个子集叫做这个关系的一个等价类。

定义 2 设 R 是集合 A 上的等价关系。与 A 中的一个元素 a 有关系的所有元素的集合叫做 a 的等价类。A 的关于 R 的等价类记作 $[a]_R$。当只有一个关系被考虑时我们将省去下标 R 并把这个等价类写作 $[a]$。

换句话说，如果 R 是集合 A 上的等价关系，元素 a 的等价类是

$$[a]_R=\{s\mid (a,s)\in R\}$$

如果 $b\in[a]_R$，b 叫做这个等价类的代表。

例 5 对于例 2 的等价关系一个整数的等价类是什么？

解 在这个等价关系中一个整数等价于它自身和它的负数。从而 $[a]=\{a,-a\}$。这个集合包含两个不同的整数，除非 $a=0$。例如，$[7]=\{7,-7\}$，$[-5]=\{-5,5\}$，$[0]=\{0\}$。■

例 6 对于模 4 同余关系，0 和 1 的等价类是什么？

解 0 的等价类包含使得 $a\equiv 0\ (\mathrm{mod}\ 4)$的所有整数 a。这个类中的整数是被 4 整除的那些整数。因此，对于这个关系 0 的等价类是

$$[0]=\{\cdots,-8,-4,0,4,8,\cdots\}$$

1 的等价类包含使得 $a\equiv 1\ (\mathrm{mod}\ 4)$ 的所有整数 a。这个类中的整数是当被 4 除时余数为 1 的那些整数。因此，对于这个关系 1 的等价类是

$$[1]=\{\cdots,-7,-3,1,5,9,\cdots\}$$ ■

在例 6 找到了 0 和 1 关于模 4 同余的等价类。用任何正整数 m 代替 4 很容易把例 6 加以推广。模 m 同余关系的等价类叫作模 m 同余类。整数 a 模 m 的同余类记作 $[a]_m$，$[a]_m=\{\cdots, a-2m, a-m, a, a+m, a+2m, \cdots\}$。例如，从例 6 得出 $[0]_4=\{\cdots, -8, -4, 0, 4, 8\cdots\}$ 和 $[1]_4=\{\cdots, -7, -3, 1, 5, 9, \cdots\}$。

6.5.4 等价类与划分

设 A 是你们学校恰好主修一个专业的学生的集合，R 是 A 上的关系，如果 x 和 y 是主修同一专业的学生，则 (x, y) 属于 R。那么正如读者可以验证的，R 是等价关系。我们可以看出 R 将 A 中的所有学生分成不相交的子集，其中每个子集包含某个特定专业的学生。例如，一个子集包含所有（只主修）计算机专业的学生，第二个子集包含所有历史专业的学生。而且这些子集是 R 的等价类。这个例于说明一个等价关系的等价类怎样把一个集合划分成不相交的非空子集。我们将在下面的讨论中把这些概念进一步精确化。

设 R 是集合 A 上的等价关系。下面的定理证明 A 的两个元素的等价类或是相等或是不相交。

定理 1 设 R 是集合 A 上的等价关系，下面的命题是等价的。

(ⅰ) aRb

(ⅱ) $[a]=[b]$

(ⅲ) $[a]\cap[b]\neq\emptyset$

证 首先证明(ⅰ)推出(ⅱ)。假设 aRb，我们将通过$[a]\subseteq[b]$和$[b]\subseteq[a]$来证明$[a]=[b]$。假设 $c\in[a]$，那么 aRc。因为 aRb 和 R 的对称性，有 bRa。又由于 R 是传递的以及 bRa 和 aRc，就得到 bRc，因而有 $c\in[b]$。这就证明了$[a]\subseteq[b]$。类似地可证明$[b]\subseteq[a]$，证明留给读者作为练习。

其次我们将证明(ⅱ)推出(ⅲ)。假设$[a]=[b]$。这就证明了$[a]\cap[b]\neq\emptyset$，因为$[a]$是非空的（由于 R 的自反性，$a\in[a]$）。

下面证明(ⅲ)推出(ⅰ)。假设$[a]\cap[b]\neq\emptyset$。那么存在元素 c 满足 $c\in[a]$和 $c\in[b]$。换句话说，aRc 和 bRc。由对称性有 cRb。再根据传递性，由 aRc 和 cRb 就有 aRb。

因为(ⅰ)推出(ⅱ)，(ⅱ)推出(ⅲ)，(ⅲ)推出(ⅰ)，三个问题(ⅰ)，(ⅱ)和(ⅲ)是等价的。 □

我们现在将显示一个等价关系怎样划分一个集合。设 R 是集合 A 上的等价关系。R 的所有等价类的并集就是 A 的全部，因为 A 的每个元素 a 都在它自己的等价类即$[a]_R$ 中。

换句话说，

$$\bigcup_{a\in A}[a]_R = A$$

此外，由定理1，这些等价类或是相等的或是不相交的，因此当$[a]\neq[b]$时，

$$[a]\cap[b]=\varnothing$$

这两个观察证明等价类构成 A 的划分，因为它们将 A 分成不相交的子集。更精确地说，集合 S 的划分是一族 S 的不相交的非空子集，且 S 就是它们的并。换句话说，一族子集 A_i，$i\in I$（其中 I 是指标集）构成 S 的划分，当且仅当

$$A_i\neq\varnothing，对于 i\in I$$
$$A_i\cap A_j=\varnothing，当 i\neq j$$

和

$$\bigcup_{i\in I}A_i=S$$

（这里符号$\bigcup_{i\in I}A_i$ 表示对于所有的 $i\in I$ 集合 A_i 的并集）。图6-13说明了集合的划分的概念。

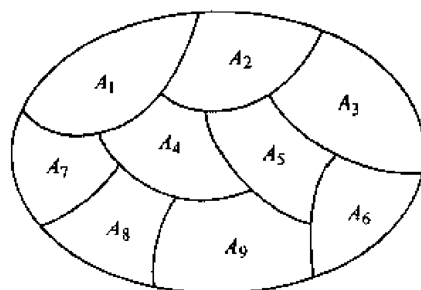


图6-13　集合的划分

例7　假设 $S=\{1,2,3,4,5,6\}$，一族集合 $A_1=\{1,2,3\}$，$A_2=\{4,5\}$ 和 $A_3=\{6\}$ 构成 S 的一个划分，因为这些集合是不相交的，且它们的并是 S。■

我们已经看到集合上等价关系的等价类构成这个集合的一个划分。这个划分中的子集就是这些等价类。反过来，可以用集合的每个划分来构成等价关系。两个元素关于这个关系是等价的，当且仅当它们在这个划分的同一子集中。

为看到这一点，假设$\{A_i|i\in I\}$是 S 的划分。设 R 是 S 上的由有序对(x,y)组成的等价关系，其中 x 和 y 属于这个划分的同一子集 A_i。为证明 R 是等价关系，我们必须证明 R 是自反的、对称的和传递的。

对于每个 $a\in S$ 有$(a,a)\in R$，因为 a 和它自己在同一子集中。因此 R 是自反的。如果$(a,b)\in R$，那么 b 和 a 在这个划分的同一子集中，因此有$(b,a)\in R$。从而 R 是对称的。如果$(a,b)\in R$ 和$(b,c)\in R$，那么 a 和 b 在这个划分的同一子集中，比如是 X，b 和 c 也在这个划分的同一子集中，比如是 Y。因为划分的子集是不相交的，并且 b 属于 X 和 Y，必有 $X=Y$。因而 a 和 c 属于这个划分的同一子集，即$(a,c)\in R$。于是 R 是传递的。

这就证明了 R 是等价关系。R 的等价类由 S 的子集构成，这些子集包含了 S 的有关系的元素，根据 R 的定义，它们就是划分的子集。定理2是对等价关系和划分之间建立的这种联系的总结。

定理2　设 R 是集合 S 上的等价关系。那么 R 的等价类构成 S 的划分。反过来，给定集合 S 的划分$\{A_i|i\in I\}$，存在着等价关系 R，它以集合 A_i，$i\in I$ 作为它的等价类。

模 m 同余类对定理2提供了一个有用的说明。存在 m 个不同的模 m 同余类，对应于当一个整数除以 m 时可能得到的 m 个不同的余数。这 m 个同余类记作$[0]_m$，$[1]_m$，…，

$[m-1]_m$ 。它们构成了整数集合的划分。

例 8　在模 4 同余产生的整数划分中的集合是什么？

解　存在 4 个同余类，对应于$[0]_4$，$[1]_4$，$[2]_4$和$[3]_4$ 。它们是集合

$$[0]_4=\{\cdots,-8,-4,0,4,8,\cdots\}$$
$$[1]_4=\{\cdots,-7,-3,1,5,9,\cdots\}$$
$$[0]_4=\{\cdots,-6,-2,2,6,10,\cdots\}$$
$$[0]_4=\{\cdots,-5,-1,3,7,11,\cdots\}$$

这些同余类是不相交的，并且每个整数恰好在它们之中的一个。换句话说，正如定理 2 所说，这些同余类构成一个划分。 ■

练习

1. 下面是$\{0,1,2,3\}$上的关系，其中哪些是等价关系？确定一个等价关系中为其他等价关系所缺少的性质。
   a) $\{(0,0),(1,1),(2,2),(3,3)\}$
   b) $\{(0,0),(0,2),(2,0),(2,2),(2,3),(3,2),(3,3)\}$
   c) $\{(0,0),(1,1),(1,2),(2,1),(2,2),(3,3)\}$
   d) $\{(0,0),(1,1),(1,3),(2,2),(2,3),(3,1),(3,2),(3,3)\}$
   e) $\{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,2),(3,3)\}$
2. 下面是所有人集合上的关系，其中哪些是等价关系？确定一个等价关系中为其他等价关系所缺少的性质。
   a) $\{(a,b)|a$ 与 b 有相同的年龄$\}$
   b) $\{(a,b)|a$ 与 b 有相同的父母$\}$
   c) $\{(a,b)|a$ 与 b 有一个相同的父亲或者一个相同的母亲$\}$
   d) $\{(a,b)|a$ 与 b 相识$\}$
   e) $\{(a,b)|a$ 与 b 说同一种语言$\}$
3. 下面是从 $\mathbf{Z}$ 到 $\mathbf{Z}$ 的所有函数集合上的关系，其中哪些是等价关系？确定一个等价关系中为其他等价关系所缺少的性质。
   a) $\{(f,g)|f(1)=g(1)\}$
   b) $\{(f,g)|f(0)=g(0)$或 $f(1)=g(1)\}$
   c) $\{(f,g)|$对所有的 $x\in\mathbf{Z},f(x)-g(x)=1\}$
   d) $\{(f,g)|$对某个 $C\in\mathbf{Z}$,对所有的 $x\in\mathbf{Z},f(x)-g(x)=C\}$
   e) $\{(f,g)|f(0)=g(1)$且 $f(1)=g(0)\}$
4. 定义 3 个在你们离散数学班中学生集合上的等价关系，要求与书中讨论的关系不同。确定关于这些等价关系的等价类。
5. 假设 A 是非空集合，f 是以 A 作为定义域的函数。设 R 是 A 上的关系，若$f(x)=f(y)$，则(x,y)属于 R。
   a) 证明 R 是A 上的等价关系。

b) R 的等价类是什么？

6. 假设 A 是非空集合，R 是 A 上的等价关系。证明存在以 A 作为定义域的函数 f，使得 $(x,y)\in R$，当且仅当 $f(x)=f(y)$。

7. 设 R 是长度至少为 3 的所有二进制串的集合上的关系，R 由对 (x,y) 构成，其中 x 和 y 在它们的前 3 位相同。证明 R 是等价关系。

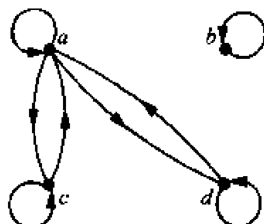
8. 设 R 是长度至少为 3 的所有二进制串的集合上的关系，R 由对 (x,y) 构成，其中 x 和 y 除了在它们的前 3 位有可能不同之外其他位都相同。证明 R 是等价关系。

9. 证明命题等价是在所有复合命题集合上等价的关系。

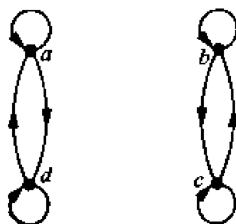
10. 设 R 是正整数的有序对集合上的关系 $((a,b),(c,d))\in R$，当且仅当 $ad=bc$。证明 R 是等价关系。

在练习 11～13 中确定具有所示有向图的关系是否为等价关系。

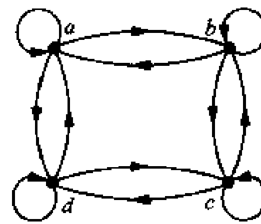
11.



12.



13.



14. 确定由下面的 0－1 矩阵表示的关系是否为等价关系。

a) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

15. 设 R 是所有二进制串的集合上的关系，sRt 当且仅当 s 和 t 包含相同个数的 1。证明 R 是等价关系。

16. 练习 1 中的等价关系的等价类是什么？

17. 练习 2 中的等价关系的等价类是什么？

18. 练习 3 中的等价关系的等价类是什么？

19. 对于练习 15 中的等价关系，二进制串 011 的等价类是什么？

20. 对于练习 7 的等价关系，下述二进制串的等价类是什么？

a) 010　　b) 1011　　c) 11111　　d) 01010101

21. 对于练习 8 的等价关系，描述练习 20 中的二进制串的等价类。

22. 当 m 是下面的整数时，什么是 $[4]_m$ 的同余类？

a) 2　　b) 3　　c) 6　　d) 8

23. 描述每一个模 6 同余类。

24. a) 对于练习 10 中的等价关系，(1, 2) 的等价类是什么？

b) 对于练习 10 中的等价关系，解释等价类的含义。

25. 下面哪些子集族是 {1, 2, 3, 4, 5, 6} 的划分？

a) {1, 2}, {2, 3, 4}, {4, 5, 6}　　b) {1}, {2, 3, 6}, {4}, {5}

c) {2, 4, 6}, {1, 3, 5}　　d) {1, 4, 5}, {2, 6}

26. 下面哪些子集族是整数集合的划分？

a) 偶数集合与奇数集合。

b) 正整数集合与负整数集合。

c) 被 3 整除的整数集合，当被 3 除时余数为 1 的整数集合，当被 3 除时余数为 2 的整数集合。

d) 小于 −100 的整数集合，绝对值不超过 100 的整数集合，大于 100 的整数集合。

e) 不能被 3 整除的整数集合，偶数集合，当被 6 除时余数为 3 的整数集合。

如果在划分 P_1 中的每个集合都是划分 P_2 中每个集合的子集，则 P_1 叫做 P_2 的加细。

27. 证明由模 6 同余类构成的划分是模 3 同余类构成的划分的加细。

28. 假设 R_1 和 R_2 是集合 A 上的等价关系，P_1 和 P_2 分别是对应于 R_1 和 R_2 的划分。证明 $R_1 \subseteq R_2$，当且仅当 P_1 是 P_2 的加细。

29. 求出在集合 $\{a,b,c,d,e\}$ 上包含关系 $\{(a,b),(a,c),(d,e)\}$ 的最小的等价关系。

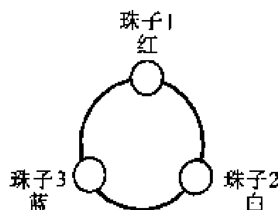
30. 假设 R_1 和 R_2 是集合 S 上的等价关系。确定下面 R_1 与 R_2 的每个组合是否一定为等价关系。

a) $R_1 \cup R_2$　　b) $R_1 \cap R_2$　　c) $R_1 \oplus R_2$

31. 考虑例 3 的等价关系，即 $R = \{(x,y) \mid x-y$ 是整数$\}$。

a) 关于这个等价关系 1 的等价类是什么？

b) 关于这个等价关系 1/2 的等价类是什么？



*32. 如例子中所显示的，在具有 3 颗珠子的手镯上每颗珠子可以是红的、白的或蓝的。如下定义手镯之间的等价关系 R：设 B_1 和 B_2 是手镯，(B_1,B_2) 属于 R，当且仅当 B_2 可以由旋转 B_1，或先旋转 B_1 然后再翻转 B_1 得到。

a) 证明 R 是等价关系。

b) R 的等价类是什么？

*33. 设 R 是 2×2 棋盘的所有涂色集合上的关系，其中 4 个方格中的每一个可以被涂成红色或蓝色。C_1 和 C_2 是被这样涂色的 2×2 棋盘，(C_1,C_2) 属于 R，当且仅当 C_2 可以由旋转 C_1，或旋转 C_1 然后再翻转 C_1 得到。

a) 证明 R 是等价关系。

b) R 的等价类是什么？

34. a) 设 R 是从 $\mathbf{Z}^+$ 到 $\mathbf{Z}^+$ 的函数的集合上的关系，且 $(f,g) \in R$，当且仅当 f 是 $\Theta(g)$（见 1.8 节）。证明 R 是等价关系。

b) 对于 a) 的等价关系描述包含 $f(n) = n^2$ 的等价类。

35. 通过列举确定 3 元素集合上的不同的等价关系个数。

36. 通过列举确定 4 元素集合上的不同的等价关系个数。

*37. 当我们构造一个关系的自反闭包的对称闭包的传递闭包时，一定得到一个等价关系吗？

*38. 当我们构造一个关系的传递闭包的自反闭包的对称闭包时，一定得到一个等价关系吗？

39. 假设我们使用定理 2 从一个等价关系 R 构造一个划分 P。如果我们再次使用定理 2 从 P

构造一个等价关系，那么结果的等价关系 R' 是什么？

40. 假设我们使用定理 2 从一个划分 P 构造一个等价关系 R，如果我们再次使用定理 2 从 R 构造一个划分，那么结果的划分 P' 是什么？
41. 设计一个算法找出包含一个给定关系的最小的等价关系。

*42. 设 $p(n)$ 表示 n 元素集合上的不同的等价关系个数（由定理 2 也是 n 元素集合的划分个数）。证明 $p(n)$ 满足递推关系 $p(n)=\sum_{j=0}^{n-1}C(n-1,j)p(n-j-1)$ 和初始条件 $p(0)=1$。（注：数 $p(n)$ 叫作贝尔数，以美国数学家 E.T. 贝尔命名。）

43. 用练习 42 求 n 元素集合上的不同的等价关系个数，其中 n 是不超过 10 的正整数。

6.6 偏序

6.6.1 引言

我们常常用关系对集合的某些元素或全体元素排序。例如，使用包含着字对 (x,y) 的关系对字排序，其中 x 按照字典顺序排在 y 的前面。使用包含着对 (x,y) 的关系安排课题，其中 x 和 y 是课题中的任务并且 x 必须在 y 开始之前完成。使用包含着对 (x,y) 的关系排序整数集合，其中 x 小于 y。当我们把所有形如 (x,x) 的对加到这些关系中时就得到了一个自反的、反对称的和传递的关系。这些都是刻划关系特征的性质，而这些关系使用它们相对的大小来排序集合的元素。

定义 1 集合 S 上的关系 R，如果它是自反的、反对称的和传递的，就称为偏序。集合 S 与偏序 R 一起叫作偏序集，记作 (S,R)。

例 1 证明"大于或等于"关系 $(\geqslant)$ 是整数集合上的偏序。

解 因为对所有整数 a 有 $a\geqslant a$，$\geqslant$ 是自反的。如果 $a\geqslant b$ 且 $b\geqslant a$，那么 $a=b$，因此 $\geqslant$ 是反对称的。最后，因为 $a\geqslant b$ 和 $b\geqslant c$ 推出 $a\geqslant c$，所以 $\geqslant$ 是传递的。从而 $\geqslant$ 是整数集合上的偏序且 $(\mathbf{Z},\geqslant)$ 是偏序集。 ■

例 2 整除关系 $|$ 是正整数集合上的偏序，因为正如 6.1 节所述，它是自反的、反对称的和传递的。我们看到 $(\mathbf{Z}^+,|)$ 是偏序集（$\mathbf{Z}^+$ 表示正整数集合）。 ■

例 3 证明包含关系 $\subseteq$ 是集合 S 的幂集上的偏序。

解 因为只要 A 是 S 的子集就有 $A\subseteq A$，$\subseteq$ 是自反的。因为 $A\subseteq B$ 和 $B\subseteq A$ 推出 $A=B$，因为它是反对称的。最后，因为 $A\subseteq B$ 和 $B\subseteq C$ 推出 $A\subseteq C$，$\subseteq$ 是传递的。因此，$\subseteq$ 是 $P(S)$ 上的偏序，且 $(P(S),\subseteq)$ 是偏序集。 ■

在一个偏序集中记号 $a\preccurlyeq b$ 表示 $(a,b)\in R$。使用这个记号是由于"小于或等于"关系是偏序关系的范例。（注意符号 $\preccurlyeq$ 用来表示任意偏序集的关系，并不仅仅是"小于或等于"关系。）记号 $a\prec b$ 表示 $a\preccurlyeq b$，但 $a\neq b$。如果 $a\prec b$ 我们说"a 小于 b"或"b 大于 a"。

当 a 与 b 是偏序集 $(S,\preccurlyeq)$ 的元素时，不一定有 $a\preccurlyeq b$ 或 $b\preccurlyeq a$。例如，在 $(P(\mathbf{Z}),\subseteq)$ 中，$\{1,2\}$ 与 $\{1,3\}$ 没有关系，反之亦然，因为没有一个集合被另一个集合包含。类似地，在 $(\mathbf{Z},|)$ 中，2 与 3 没关系，3 与 2 也没关系，因为 $2\nmid 3$ 且 $3\nmid 2$。由此得到下面的定义。

定义 2　偏序集$(S, \preccurlyeq)$的元素 a 和 b 叫作可比的，如果 $a \preccurlyeq b$ 或 $b \preccurlyeq a$。当 a 和 b 是 S 的元素并且既没有 $a \preccurlyeq b$，也没有 $b \preccurlyeq a$，则称 a 与 b 是不可比的。

例 4　在偏序集$(\mathbf{Z}^+, |)$中整数 3 和 9 是可比的吗？5 和 7 是可比的吗？

解　整数 3 和 9 是可比的，因为 $3|9$。整数 5 和 7 是不可比的，因为 $5 \nmid 7$ 且 $7 \nmid 5$。 ■

用形容词"部分的（偏的）"描述偏序是由于一些对元素可能是不可比的。当集合中的每对元素都可比时，这个关系叫作全序。

定义 3　如果$(S, \preccurlyeq)$是偏序集，且 S 的每对元素都是可比的，则 S 叫作全序集或线序集，且$\preccurlyeq$叫作全序或线序。一个全序集也叫作链。

例 5　偏序集$(\mathbf{Z}, \leqslant)$是全序集，因为只要 a 和 b 是整数就有 $a \leqslant b$ 或 $b \leqslant a$。 ■

例 6　偏序集$(\mathbf{Z}^+, |)$不是全序集，因为它包含着不可比的元素，例如 5 和 7。 ■

在第 3 章我们注意到$(\mathbf{Z}^+, \leqslant)$是良序的，其中$\leqslant$是通常的"小于或等于"关系。我们现在定义良序集。

定义 4　对于偏序集$(S, \preccurlyeq)$，如果$\preccurlyeq$是全序，并且 S 的每个非空子集都有一个最小元素，就称它为良序集。

例 7　正整数的有序对的集合 $\mathbf{Z}^+ \times \mathbf{Z}^+$ 与关系$\preccurlyeq$构成良序集。对于(a_1, a_2)和(b_1, b_2)，如果 $a_1 < b_1$，或如果 $a_1 = b_1$ 且 $a_2 \leqslant b_2$（字典顺序），则$(a_1, a_2) \preccurlyeq (b_1, b_2)$。有关的验证留作节后的练习。集合 $\mathbf{Z}$ 与通常的$\leqslant$序不是良序的，因为负整数集合是 $\mathbf{Z}$ 的子集，但没有最小元素。 ■

6.6.2　字典顺序

字典中的字按照字母顺序或字典顺序排列，字典顺序是以字母表中的字母顺序为基础的。这是从一个集合上的偏序构造一个集合上的串的序的特殊情况。我们将说明这种构造在一个偏序集上是怎样做的。

首先，我们将说明怎样在两个偏序集$(A_1, \preccurlyeq_1)$和$(A_2, \preccurlyeq_2)$的笛卡尔积上构造一个偏序。在 $A_1 \times A_2$ 上的字典顺序定义如下：如果第一个对的第一个元素（在 A_1 中）小于第二个对的第一个元素，或者第一个元素相等，但是第一个对的第二个元素（在 A_2 中）小于第二个对的第二个元素，那么第一个对小于第二个对。换句话说，(a_1, a_2)小于(b_1, b_2)，即

$$(a_1, a_2) \prec (b_1, b_2)$$

或者 $a_1 \prec_1 b_1$，或者 $a_1 = b_1$ 且 $a_2 \prec_2 b_2$。

把相等加到 $A \times B$ 上的序$\prec$就得到偏序$\preccurlyeq$。这个验证留作练习。

例 8　确定在偏序集$(\mathbf{Z} \times \mathbf{Z}, \preccurlyeq)$中是否有$(3,5) \prec (4,8)$？$(3,8) \prec (4,5)$？$(4,9) \prec (4,11)$？这里的$\preccurlyeq$是从 $\mathbf{Z}$ 上通常的$\leqslant$关系构造的字典顺序。

解　因为 $3 < 4$，因此$(3,5) \prec (4,8)$且$(3,8) \prec (4,5)$。因为$(4,9)$与$(4,11)$的第一元素相同，但是 $9 < 11$，我们有$(4,9) \prec (4,11)$。 ■

图6-14明显地显示了$\mathbf{Z}^+\times\mathbf{Z}^+$中比(3,4)小的有序对的集合。

可以在n个偏序集$(A_1,\preccurlyeq_1)$，$(A_2,\preccurlyeq_2)$，…，$(A_n,\preccurlyeq_n)$的笛卡儿积上定义字典顺序。如下定义$A_1\times A_2\times\cdots\times A_n$上的偏序：如果$a_1\prec b_1$，或者存在整数$i>0$使得$a_1=b_1$，…，$a_i=b_i$，且$a_{i+1}\prec b_{i+1}$，那么

$$(a_1,a_2,\cdots,a_n)\prec(b_1,b_2,\cdots,b_n)$$

换句话说，如果在两个n元组不同元素出现的第一位置上第一个n元组的元素小于第二个n元组的元素，那么第一个n元组小于第二个n元组。

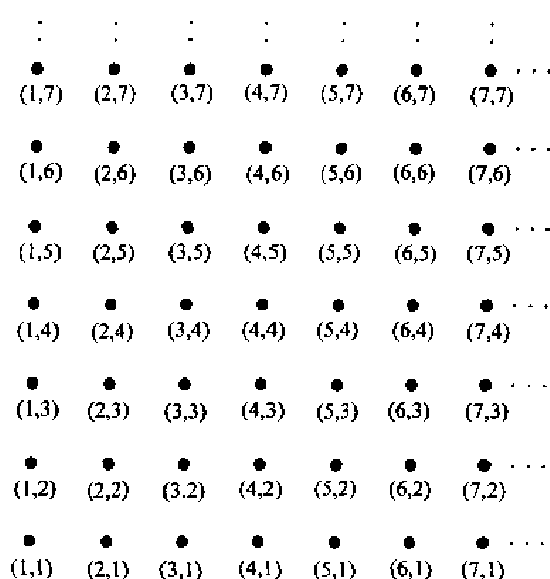


图6-14 按照字典顺序比(3,4)小的有序对

例9 注意$(1,2,3,5)\prec(1,2,4,3)$，因为这些4元组的前两位相同，但是第一个4元组的第三位3小于第二个4元组的第三位4（这里的4元组上的字典顺序是整数集合上的通常的“小于或等于”关系导出的字典顺序。）■

我们现在可以定义串上的字典顺序。考虑偏序集S上的串$a_1a_2\cdots a_m$和$b_1b_2\cdots b_n$假定这些串不相等。设t是m，n中较小的数。定义字典顺序如下：$a_1a_2\cdots a_m$小于$b_1b_2\cdots b_n$，当且仅当

$$(a_1,a_2,\cdots,a_t)\prec(b_1,b_2,\cdots,b_t)\text{或者}$$
$$(a_1,a_2,\cdots,a_t)=(b_1,b_2,\cdots,b_t)\text{并且 }m<n$$

其中这个不等式中的$\prec$表示S^t中的字典顺序。换句话说，为确定两个不同串的序，较长的串被切到较短的串的长度t，即$t=\min(m,n)$。然后使用S^t上的字典顺序比较每个串的前t位组成的t元组。如果对应于第一个串的t元组小于第二个串的t元组，或者这两个t元组相等，但是第二个串更长，那么第一个串小于第二个串。这是偏序的验证留给读者作为练习。

例10 考虑小写英语字母的串构成的集合。使用在字母表中的字母序可以构造在串的集合上的字典顺序。如果在两个串出现不同字母的首位，第一个串的字母在第二个串的同位字母的前边，或者如果第一个串和第二个串在所有的位都相同，但是第二个串有更多的字母，那么，第一个串小于第二个串。这个序和字典使用的序相同。例如，

discreet$\prec$discrete

因为这两个串在第7位首次出现不同字母，并且e$\prec$t。

discreet$\prec$discreetness

因为这两个串前8个字母相同，但是第二个串更长。此外，

discrete$\prec$discretion

因为

discrete≺discreti ■

6.6.3　哈斯图

在有穷偏序集的有向图中有许多边可以不必显示出来，因为它们是必须存在的。例如，考虑在集合{1,2,3,4}上的偏序{(*a*,*b*)|*a*≤*b*}的有向图，该图给在图6-15 a)。因为这个关系是偏序，它是自反的并且有向图在所有的顶点都有环。因此，我们不必显示这些环，因为它们是必须出现的。在图6-15 b）中没有显示这些环。由于一个偏序是传递的，我们不必显示那些由于传递性而必须出现的边。例如，在图6-15 c）中没有显示边(1,3),(1,4),和(2,4)，因为它们是必须出现的。如果我们假设所有边的方向是向上的（如它们在图中所示），我们不必显示边的方向；图6-15 c）没有显示方向。

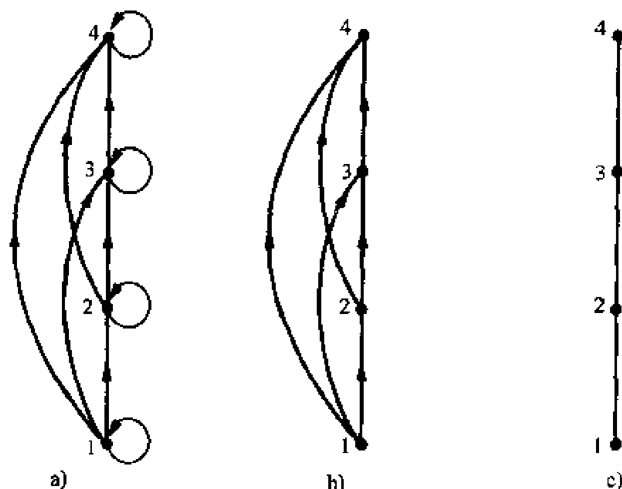


图6-15　构造关于({1,2,3,4},≤)的哈斯图

一般说来，我们可以使用下面的过程表示一个有穷集上的偏序。从这个关系的有向图开始。由于偏序是自反的，每个顶点有一个环，移走这些环。移走所有由于传递性出现的边，因为偏序是传递的，这些边是必须出现的。例如，如果(*a*,*b*)和(*b*,*c*)在偏序中，移走边(*a*,*c*)，因为它必须要出现。此外，如果(*c*,*d*)也在偏序中，移走边(*a*,*d*)，因为它也是必须出现的。最后，排列每条边使得它的始点在终点的下面（正如在纸上所画的）。移走所有有向边上的箭头，因为所有的边向上指向它们的终点（只有和偏序集的覆盖关系中的对所对应的边留下来。见练习20前面的说明）。

这些步是严格定义的，并且对于一个有穷偏序集只有有限步需要执行。当所有的步被执行以后，就得到一个包含着足够的表示偏序信息的图。这个图叫作哈斯图，它是用20世纪德国数学家赫尔姆·哈斯[㊀]的名字命名的。

㊀ 赫尔姆·哈斯（Helmut Hasse，1898—1979）　哈斯生于卡斯尔。他高中毕业后在德国海军服役。他于1918年在哥廷根大学开始大学学习，并于1920年到马尔堡大学的数论专家科特·亨塞尔指导下学习。在这段时间，哈斯对代数数论作出了基础性的贡献。他接替了亨塞尔在马尔堡大学的工作，后来于1934年成为著名的哥廷根数学研究所的所长，并且在1950年受聘于汉堡大学。哈斯作为著名的德国数学期刊《Crelle学报》的编辑工作了50年。当纳粹迫使亨塞尔辞职时他在1936年承担了主编的工作。在第二次世界大战时期，哈斯为德国海军从事应用数学研究。他以讲课的清晰和个人风格而著称，并且献身于数论和他的学生。（哈斯由于与纳粹党的联系而受到非议。调查证明他是强烈的德国民族主义者，但不是狂热的纳粹分子。）

例 11　画出表示{1,2,3,4,6,8,12}上的偏序{(a,b)|a 整除 b}的哈斯图。

解　由这个偏序的有向图开始，如图 6-16 a）所示。移走所有的环，如图 6-16 b）所示。然后删除所有由传递性导出的边。这些边是(1,4),(1,6),(1,8),(1,12),(2,8),(2,12)和(3,12)。排列所有的边使得方向向上，并且删除所有的箭头得到哈斯图。结果的哈斯图显示在图 6-16 c)。■

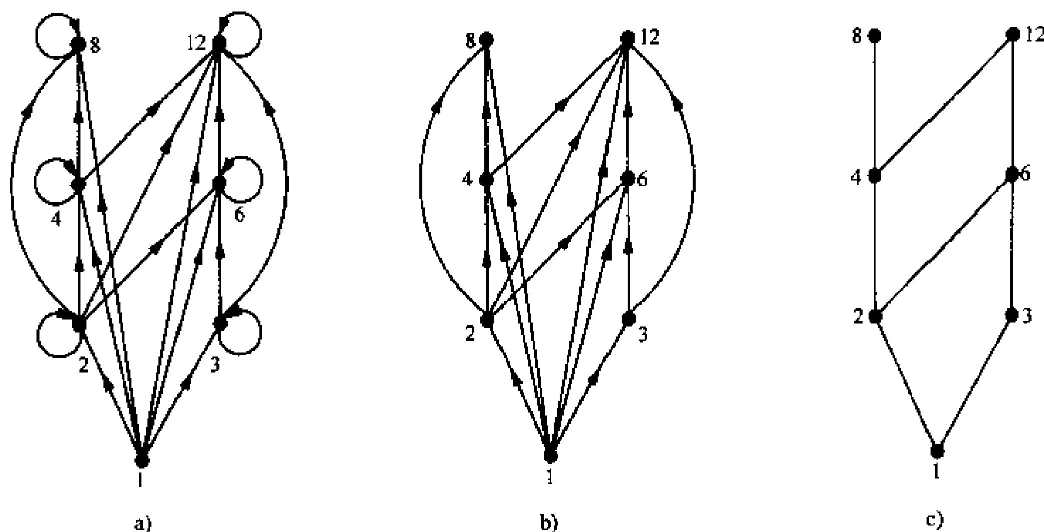


图 6-16　构造({1,2,3,4,6,8,12},|)上的哈斯图

例 12　画出幂集 $P(S)$ 上的偏序 $\{(A,B)\mid A\subseteq B\}$ 的哈斯图，其中 $S=\{a,b,c\}$。

解　关于这个偏序的哈斯图是由相关的有向图得到的，先删除所有的环和所有由传递性产生的边，即 $(\emptyset,\{a,b\})$，$(\emptyset,\{a,c\})$，$(\emptyset,\{b,c\})$，$(\emptyset,\{a,b,c\})$，$(\{a\},\{a,b,c\})$，$(\{b\},\{a,b,c\})$ 和 $(\{c\},\{a,b,c\})$。最后，使所有的边方向向上，并删除箭头。结果的哈斯图如图 6-17 所示。■

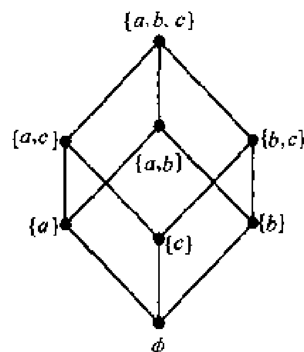


图 6-17　$(P(\{a,b,c\}),\subseteq)$的哈斯图

6.6.4　极大元素与极小元素

有某种特殊性质的偏序集的元素对许多应用是很重要的。偏序集的一个元素叫作极大的，如果它不小于这个偏序集的任何其他元素。即 a 是偏序集 $(S,\preccurlyeq)$ 的*极大元素*，如果不存在 $b\in S$ 使得 $a\prec b$。类似地，偏序集的一个元素叫作*极小的*，如果它不大于这个偏序集的任何其他元素。即 a 是偏序集 $(S,\preccurlyeq)$ 的*极小元素*，如果不存在 $b\in S$ 使得 $b\prec a$。使用哈斯图很容易识别极大元与极小元素。它们是图中的“顶”元素与“底”元素。

例 13　偏序集({2,4,5,10,12,20,25},|)的哪些元素是极大元素，哪些是极小元素？

解　图 6-18 关于这个偏序集的哈斯图显示了极大元素是 12，20 和 25，极小元素是 2 和 5。通过这个例子可以看出，一个偏序集可以有多于一个的极大元素和多于一个的极小元

素。■

有时在偏序集中存在一个元素大于每个其他的元素。这样的元素叫作最大元素。即 a 是偏序集 $(S, \preccurlyeq)$ 的最大元素，如果对所有的 $b \in S$ 有 $b \preccurlyeq a$。当最大元素存在时它是唯一的［见节末练习 32 a)］。类似地，一个元素叫作最小元素，如果它小于偏序集的所有其他的元素。即 a 是偏序集 $(S, \preccurlyeq)$ 的最小元素，如果对所有的 $b \in S$ 有 $a \preccurlyeq b$。当最小元素存在时它也是唯一的［见节末练习 32 b)］。

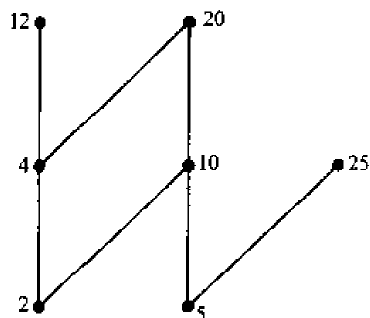


图 6-18　偏序集的哈斯图

例 14　确定图 6-19 的每个哈斯图表示的偏序集是否有最大元素和最小元素。

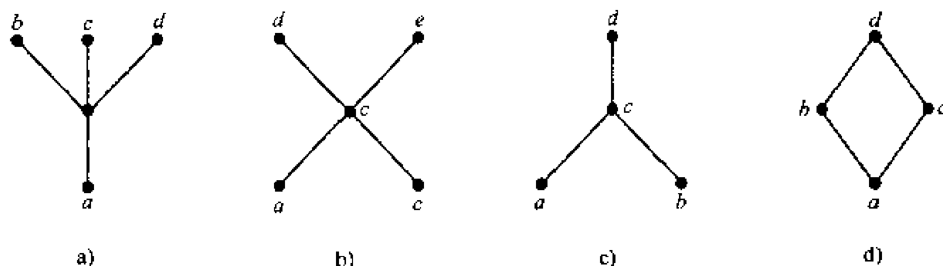


图 6-19　四个偏序集的哈斯图

解　哈斯图 a) 的偏序集的最小元素是 a。这个偏序集没有最大元素。哈斯图 b) 的偏序集既没有最小元素也没有最大元素。哈斯图 c) 的偏序集没有最小元素。它的最大元素是 d。哈斯图 d) 的偏序集有最小元素 a 和最大元素 d。■

例 15　设 S 是集合。确定偏序集$(P(S), \subseteq)$中是否存在最大元素与最小元素。

解　最小元素是空集，因为对于 S 的任何子集 T 有 $\emptyset \subseteq T$。集合 S 是这个偏序集的最大元素，因为只要 T 是 S 的子集，就有 $T \subseteq S$。■

例 16　在偏序集$(\mathbf{Z}^+, |)$中是否存在最大元素和最小元素。

解　1 是最小元素，因为只要 n 是正整数，就有 $1 | n$。因为没有被所有正整数整除的整数，所以不存在最大元素。■

有时候可以找到一个元素大于偏序集$(S, \preccurlyeq)$的子集 A 中所有的元素。如果 u 是 S 的元素，使得对所有的元素 $a \in A$ 有 $a \preccurlyeq u$，那么 u 叫作 A 的一个上界。类似地，也可能存在一个元素小于 A 中的所有其他元素。如果 l 是 S 的一个元素，使得对所有的元素 $a \in A$ 有 $l \preccurlyeq a$，那么 l 叫作 A 的一个下界。

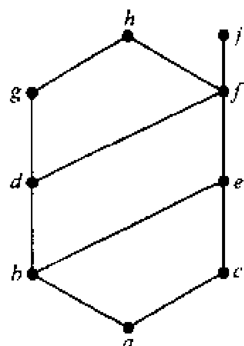


图 6-20　偏序集的哈斯图

例 17　找出在图 6-20 所示哈斯图的偏序集的子集$\{a, b, c\}$，$\{j, h\}$和$\{a, c, d, f\}$的下界和上界。

解　$\{a, b, c\}$的上界是 e，f，j 和 h，它的唯一的下界

是 a。$\{j,h\}$没有上界，它的下界是 a，b，c，d，e 和 f。$\{a,c,d,f\}$的上界是 f，h 和 j，它的下界是 a。 ■

元素 x 叫作子集 A 的最小上界，如果 x 是一个上界并且它小于 A 的每个其他的上界。因为如果这样的元素存在，只存在一个，称这个元素为最小上界是有意义的（见节末练习 34 a)）。即如果只要 $a\in A$ 就有 $a\preccurlyeq x$，并且只要 z 是 A 的上界，就有 $x\preccurlyeq z$，x 就是 A 的最小上界。类似地，如果 y 是 A 的下界，并且只要 z 是 A 的下界，就有 $z\preccurlyeq y$，y 就称为 A 的最大下界。A 的最大下界如果存在也是唯一的（见节末的练习 34 b)）。一个子集 A 的最大下界和最小上界分别记作 glb(A)和 lub(A)。

例 18 在图 6-20 所示偏序集中如果$\{b,d,g\}$的最大下界和最小上界存在，求出这个最大下界和最小上界。

解 $\{b,d,g\}$的上界是 g 和 h。因为 $g\prec h$，g 是最小上界。$\{b,d,g\}$的下界是 a 和 b。因为 $a\preccurlyeq b$，b 是最大下界。 ■

例 19 在偏序集$(\mathbf{Z}^+,|)$中如果集合$\{3,9,12\}$和$\{1,2,4,5,10\}$的最大下界和最小上界存在，求出这些最大下界和最小上界。

解 如果 3，9，12 被一个整数整除，那么这个整数就是$\{3,9,12\}$的下界。这样的整数只有 1 和 3。因为 $1|3$，3 是$\{3,9,12\}$的最大下界。集合$\{1,2,4,5,10\}$关系到$|$的下界只有 1，因此 1 是$\{1,2,4,5,10\}$的最大下界。

一个整数是$\{3,9,12\}$的上界，当且仅当它被 3，9 和 12 整除。具有这种性质的整数就是那些被 3，9 和 12 的最小公倍数 36 整除的整数。因此，36 是$\{3,9,12\}$的最小上界。一个正整数是集合$\{1,2,4,5,10\}$的上界，当且仅当它被 1，2，4，5 和 10 整除。具有这种性质的整数就是被这些整数的最小公倍数 20 整除的整数。因此，20 是$\{1,2,4,5,10\}$的最小上界。 ■

6.6.5 格

如果一个偏序集的每对元素都有最小上界和最大下界，就称这个偏序集为格。格有许多特殊的性质。此外，格有许多不同的应用，如在信息流的模型中，并且格在布尔代数中起了重要的作用。

例 20 确定图 6-21 的每个哈斯图表示的偏序集是否是格。

解 在 a）和 c）中的哈斯图表示的偏序集是格，因为在每个偏序集中每对元素都有最小上界和最大下界，读者应该能验证这一点。另一方面，b）所示的哈斯图的偏序集不是格，

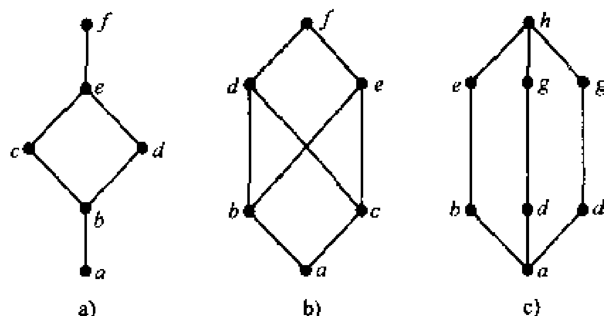


图 6-21 三个偏序集的哈斯图

因为元素 b 和 c 没有最小上界。为此只要注意到 d，e 和 f 中每一个都是上界，但这 3 个元素的任何一个关于这个偏序集中的序都不大于其他 2 个。■

例 21 偏序集$(\mathbf{Z}^+,|)$是格吗？

解 设 a 和 b 是两个正整数。这两个整数的最小上界和最大下界分别是它们的最小公倍数和最大公约数，读者应能验证这一点。因此这个偏序集是格。■

例 22 确定偏序集$(\{1,2,3,4,5\},|)$和$(\{1,2,4,8,16\},|)$是否为格。

解 因为 2 和 3 在$(\{1,2,3,4,5\},|)$中没有上界，它们当然没有最小上界。因此第一个偏序集不是格。

第二个偏序集的每两个元素都有最小上界和最大下界。在这个偏序集中两个元素的最小上界是它们中间较大的元素，而两个元素的最大下界是它们中间较小的元素，读者应能验证这一点。因此第二个偏序集是格。■

例 23 确定$(P(S),\subseteq)$是否是格，其中 S 是集合。

解 设 A 和 B 是 S 的两个子集。A 和 B 的最小上界和最大下界分别是$A\cup B$ 和 $A\cap B$，正如读者可以证明的。因此$(P(S),\subseteq)$是格。■

例 24 信息流的格模型。在许多设置中从一个人或计算机程序到另一个人或计算机程序的信息流要受到限制，这可以通过安全权限来实现。我们可以使用格的模型来表示不同的信息流策略。例如，一个通用的信息流策略是用于政府或军事系统中的多级安全策略。为每组信息分配一个安全级别，并且每个安全级别用一个对(A,C)表示，其中 A 是权限级别，C 是种类。然后允许人和计算机程序从一个被特别限制的安全类的集合中访问信息。

在美国政府中使用的典型的权限级别是不保密(0)、秘密(1)、机密(2)和绝密(3)。在安全级别中使用的种类是一个集合的子集，这个集合含有与一个特定行业领域相关的所有的分部，每个分部表示一个指定的课题领域。例如，如果分部的集合是{间谍，鼹鼠，双重间谍}，那么存在 8 个不同的种类，分部集合有 8 个子集，对于每个子集有一类，例如{间谍，鼹鼠}。

我们可以对安全类排序，规定$(A_1,C_1)\preccurlyeq(A_2,C_2)$，当且仅当 $A_1\leqslant A_2$ 和 $C_1\subseteq C_2$。信息允许从安全类(A_1,C_1)流向安全类(A_2,C_2)，当且仅当$(A_1,C_1)\preccurlyeq(A_2,C_2)$。例如，信息允许从安全类（机密，{间谍，鼹鼠}）流向安全类（绝密，{间谍，鼹鼠，双重间谍}），反之，信息不允许从安全类（绝密，{间谍，鼹鼠}）流向安全类（机密，{间谍，鼹鼠，双重间谍}）或（绝密，{间谍}）。

我们留给读者（见节末练习 40）证明所有安全类的集合与在这个例子中所定义的序构成一个格。■

6.6.6 拓扑排序

假设一个项目由 20 个任务构成。某些任务只能在其他任务结束之后完成。怎么能找到关于这些任务的顺序？为了对这个问题构造模型，我们建立一个任务集合上的偏序，使得 $a\prec b$，当且仅当 a 和 b 是任务且直到 a 结束后 b 才能开始。为安排好这个项目，

需要得出与这个偏序相容的所有 20 个任务的顺序。我们将说明怎样可以做到这一点。

我们从定义开始。说一个全序$\preccurlyeq$同偏序 R 是相容的，如果只要 $a\,R\,b$ 就有$a\preccurlyeq b$。从一个偏序构造一个相容的全序叫作**拓扑排序**。我们需要使用下面的引理。

引理 1 每个有穷非空偏序集$(S,\preccurlyeq)$都有极小元素。

证 选择 S 的一个元素 a_0。如果 a_0 不是极小元素，那么存在元素 a_1 满足 $a_1\prec a_0$。如果 a_1 不是极小元素，那么存在元素 a_2 满足 $a_2\prec a_1$。继续这一过程，使得如果 a_n 不是极小元素，那么存在元素 a_{n+1}满足 $a_{n+1}\prec a_n$。因为在这个偏序集只有有穷个元素，这个过程一定结束，并且具有极小元素 a_n。 □

将要描述的拓扑排序算法对任何有穷非空偏序集都有效。为在偏序集$(A,\preccurlyeq)$上定义一个全序，首先选择一个极小元素 a_1；由引理 1 这样的元素存在。接着，正如读者应该验证的，$(A-\{a_1\},\preccurlyeq)$也是一个偏序集。如果它是非空的，选择这个偏序集的一个极小元素 a_2。然后再取走 a_2，如果还有其他的元素留下来，在 $A-\{a_1,a_2\}$中选择一个极小元素 a_3。继续这个过程，只要还有元素留下来，就在 $A-\{a_1,a_2,\cdots a_k\}$中选择极小元素 a_{k+1}。

因为 A 是有穷集，这个过程一定终止。最终产生一个元素序列 a_1，a_2，…，a_n。所需要的全序定义为

$$a_1\preccurlyeq a_2\preccurlyeq\cdots\preccurlyeq a_n$$

这个全序是与初始的偏序相容的。为看出这一点，如果在初始的偏序中 $b\prec c$，c 在算法的某个阶段被选择为极小元素，这时 b 已经被移出，否则 c 就不会是极小元素。关于这个拓扑排序算法的伪码给在算法 1 中。

算法 1 拓扑排序

```
procedure topological sort (S: 有穷偏序集)
k := 1
while S≠∅
begin
    a_k = S 的极小元素 {由引理 1 这样的元素一定存在}
    S := S - {a_k}
    k := k + 1
end {a_1, a_2, ⋯ a_n 是 S 的相容的全序}
```

例 25 找出对于偏序集$(\{1,2,4,5,12,20\},|)$的一个相容的全序。

解 第一步是选择一个极小元素。这个元素一个是 1，因为它是唯一的极小元素。下一步选择$(\{2,4,5,12,20\},|)$的一个极小元素。在这个偏序集中有两个极小元素，即 2 和 5。我们选择 5。剩下的元素是$\{2,4,12,20\}$。在这一步的唯一极小元素是 2。下一步选择 4，因为它是$(\{4,12,20\},|)$的唯一极小元素。因为 12 和 20 都是$(\{12,20\},|)$的极小元素，下一步选哪一个都可以。我们选 20，只剩下 12 作为最后的元素。这产生了全序

$$1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12$$

这个排序算法所使用的步骤给在图 6-22 中。 ■

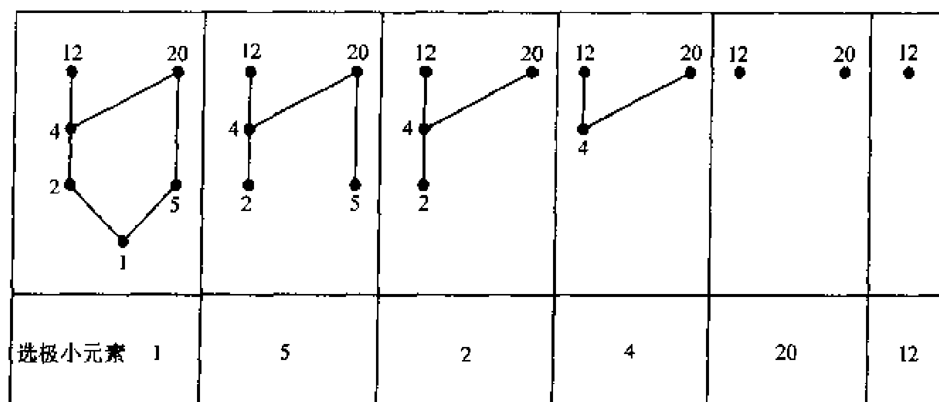


图 6-22　({1,2,4,5,12,20},|)的拓扑排序

在项目的安排中常用拓扑排序。

例 26　一个计算机公司的开发项目需要完成 7 个任务。其中的某些任务只能在其他任务结束后才能开始。考虑如下建立任务上的偏序，如果任务 Y 在 X 结束后才能开始，则任务 $X \prec$ 任务 Y。这 7 个任务关于这个偏序的哈斯图给在图 6-23 中，求一个全序使得可以按照这个全序执行这些任务以完成这个项目。

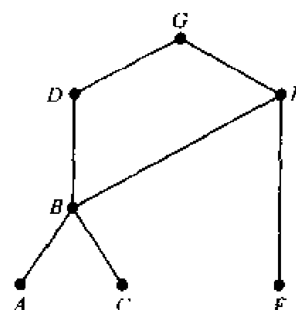


图 6-23　关于 7 个任务的哈斯图

解　可以通过执行一个拓扑排序得到 7 个任务的排序。排序的步骤显示在图 6-24 中。这个排序的结果，$A \prec C \prec B \prec E \prec F \prec D \prec G$，给出了关于任务的一种可能的次序。 ■

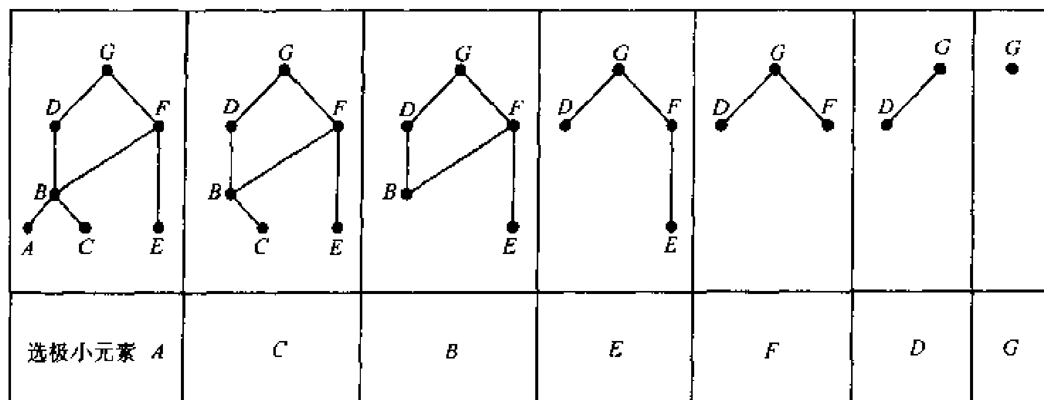


图 6-24　任务的拓扑排序

练习

1. 下面哪些集合是偏序集？

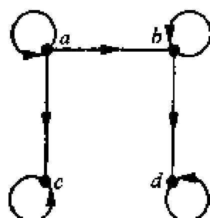
a) $(\mathbf{Z}, =)$ b) $(\mathbf{Z}, \neq)$ c) $(\mathbf{Z}, \geqslant)$ d) $(\mathbf{Z}, \nmid)$

2. 确定由下面的 0－1 矩阵表示的关系是否为偏序。

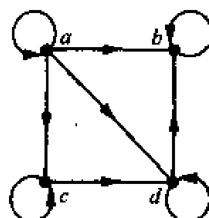
a) $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ b) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ c) $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$

在练习 3～5 确定有向图所表示的关系是否为偏序。

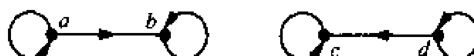
3.



4.



5.



6. 设(S,R)是偏序集。证明(S,R^{-1})也是偏序集，其中 R^{-1}是 R 的逆。偏序集(S,R^{-1})叫作(S,R)的对偶。
7. 求下面偏序集的对偶。

a) $(\{0,1,2\},\leqslant)$ b) $(\mathbf{Z},\geqslant)$ c) $(P(\mathbf{Z}),\supseteq)$ d) $(\mathbf{Z}^+,|)$

8. 在偏序集$(\mathbf{Z}^+,|)$中下面哪对元素是可比的？

a) 5，15 b) 6，9 c) 8，16 d) 7，7

9. 在下面偏序集中找出两个不可比的元素。

a) $(P(\{0,1,2\}),\subseteq)$

b) $(\{1,2,4,6,8\},|)$

10. 设 $S=\{1,2,3,4\}$。考虑基于通常小于关系的字典顺序。

a) 找出在 $S\times S$ 中所有小于$(2,3)$的对。

b) 找出在 $S\times S$ 中所有大于$(3,1)$的对。

c) 画出偏序集$(S\times S,\preccurlyeq)$的哈斯图。

11. 找出下面的 n 元组的字典顺序。

a) $(1,1,2),(1,2,1)$

b) $(0,1,2,3),(0,1,3,2)$

c) $(1,0,1,0,1),(0,1,1,1,0)$

12. 找出下面小写英语字母串的字典顺序。

a) quack，quick，quicksilver，quicksand，quacking

b) open，opener，opera，operand，opened

c) zoo，zero，zoom，zoology，zoological

13. 找出二进制串 0，01，11，001，010，011，0001 和 0101 的基于 $0<1$ 的字典顺序。
14. 画出$\{0,1,2,3,4,5\}$上的“大于或等于”关系的哈斯图。
15. 画出在下述集合上的整除关系的哈斯图。

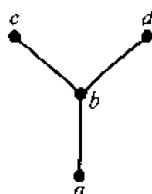
a) $\{1,2,3,4,5,6,7,8\}$ b) $\{1,2,3,5,7,11,13\}$

c) {1,2,3,6,12,24,36,48}　　d) {1,2,4,8,16,32,64}

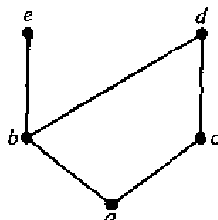
16. 画出在集合 $P(S)$上包含关系的哈斯图，其中 $S=\{a,b,c,d\}$。

在练习 17～19 列出具有所示哈斯图的偏序中的所有的有序对。

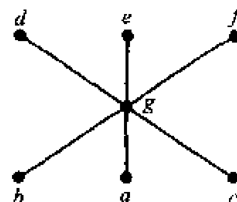
17.



18.



19.



设$(S,\preccurlyeq)$是偏序集。我们说一个元素 $y\in S$ 覆盖元素 $x\in S$，如果 $x\prec y$ 并且没有元素 $z\in S$ 使得 $x\prec z\prec y$。使得 y 覆盖 x 的对(x,y)的集合叫做$(S,\preccurlyeq)$的*覆盖关系*。

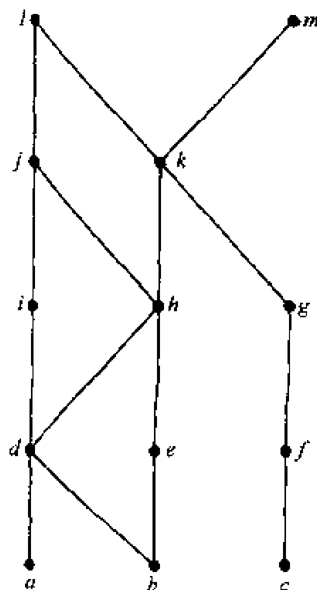
20. 什么是{1,2,3,4,6,12}上的偏序$\{(a,b)\mid a$ 整除 $b\}$的覆盖关系？

21. 什么是 S 的幂集上的偏序$\{(A,B)\mid A\subseteq B\}$的覆盖关系？其中 $S=\{a,b,c\}$。

22. 证明有序对(x,y)属于有穷偏序集$(S,\preccurlyeq)$的覆盖关系，当且仅当 x 小于 y，并且在这个偏序集的哈斯图中存在一条连接 x 和 y 的边。

23. 证明一个有穷偏序集可以从它的覆盖关系重新构造出来。

24. 对于由右边的哈斯图表示的偏序回答下列问题。



a) 求极大元素。

b) 求极小元素。

c) 存在最大元素吗？

d) 存在最小元素吗？

e) 求$\{a,b,c\}$的所有上界。

f) 如果存在的话，求$\{a,b,c\}$的最小上界。

g) 求$\{f,g,h\}$的所有下界。

h) 如果存在的话，求$\{f,g,h\}$的最大下界。

25. 对偏序集({3,5,9,15,24,45},|)回答下列问题。

a) 求极大元素。

b) 求极小元素。

c) 存在最大元素吗？

d) 存在最小元素吗？

e) 找出{3,5}的所有上界。

f) 如果存在的话，求{3,5}的最小上界。

g) 求{15,45}的所有下界。

h) 如果存在的话，求{15,45}的最大下界。

26. 对偏序集({2,4,6,9,12,18,27,36,48,60,72},|)回答下列问题。

a) 找出极大元素。

b) 找出极小元素。

c）存在最大元素吗？

d）存在最小元素吗？

e）找出{2,9}的所有上界。

f）如果存在，找出{2,9}的最小上界。

g）找出{60,72}的所有下界。

h）如果存在，找出{60,72}的最大下界。

27. 对偏序集({{1},{2},{4},{1,2},{1,4},{2,4},{3,4},{1,3,4},{2,3,4}},⊆)回答下列问题。

a）求极大元素。

b）求极小元素。

c）存在最大元素吗？

d）存在最小元素吗？

e）求{{2},{4}}的所有上界。

f）如果存在的话，求{{2},{4}}的最小上界。

g）求{{1,3,4},{2,3,4}}的所有下界。

h）如果存在的话，求{{1,3,4},{2,3,4}}的最大下界。

28. 给出满足下述性质的偏序集。

a）有一个极小元素但没有极大元素。

b）有一个极大元素但没有极小元素。

c）既没有极大元素也没有极小元素。

29. 证明字典顺序是两个偏序集的笛卡儿积上的偏序。

30. 证明字典顺序是一个偏序集的串的集合上的偏序。

31. 假设$(S,\preccurlyeq_1)$和$(T,\preccurlyeq_2)$是偏序集。证明$(S\times T,\preccurlyeq)$也是偏序集，其中$(s,t)\preccurlyeq(u,v)$，当且仅当$s\preccurlyeq_1 u$且$t\preccurlyeq_2 v$。

32. a）如果偏序集存在最大元素，证明恰好存在一个最大元素。

b）如果偏序集存在最小元素，证明恰好存在一个最小元素。

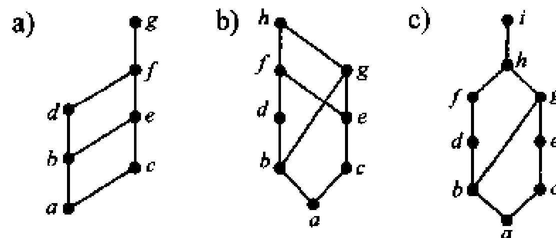
33. a）证明在一个具有最大元素的偏序集中恰好存在一个极大元素。

b）证明在一个具有最小元素的偏序集中恰好存在一个极小元素。

34. a）证明如果偏序集的子集存在最小上界的话，则是唯一的。

b）证明如果偏序集的子集存在最大下界的话，则是唯一的。

35. 确定具有下面哈斯图的偏序集是否为格。



36. 确定下面的偏序集是否为格。

a) $(\{1,3,6,9,12\},|)$

b) $(\{1,5,25,125\},|)$

c) $(\mathbf{Z},\geqslant)$

d) $(P(S),\supseteq)$，其中 $P(S)$是集合 S 的幂集

37. 证明一个格的每个有限非空子集有最小上界和最大下界。

38. 证明如果偏序集(S,R)是格，那么对偶偏序集(S,R^{-1})也是格。

39. 在一个公司里用信息流的格模型控制敏感信息，这些信息具有由有序对 (A, C) 表示的安全类别。这里 A 是权限级别，这种权限级别可以是非私有的 (0)、私有的 (1)、受限制的 (2) 或注册的 (3)。种类 C 是所有项目集合 {猎豹，黑斑羚，美洲狮} 的子集。(在公司里常常使用动物的名字作为项目的代码名字。)

   a) 信息允许从 (私有的，{猎豹，美洲狮}) 流向 (受限制的，{美洲狮}) 吗？

   b) 信息允许从 (受限制的，{猎豹}) 流向 (注册的，{猎豹，黑斑羚}) 吗？

   c) 信息从 (私有的，{猎豹，美洲狮}) 允许流向哪个类？

   d) 信息允许从哪个类流向安全类 (受限制的，{黑斑羚，美洲狮})？

40. 证明安全类别(A,C)的集合 S 是一个格，其中 A 是表示权限级别的正整数，C 是种类的有穷集的子集，具有$(A_1,C_1)\preccurlyeq(A_2,C_2)$，当且仅当 $A_1\leqslant A_2$ 且 $C_1\subseteq C_2$。[提示：首先证明$(S,\preccurlyeq)$是一个偏序集，然后证明(A_1,C_1)和(A_2,C_2)的最小上界和最大下界分别是$(\max(A_1,A_2), C_1\cup C_2)$和$(\min(A_1,A_2),C_1\cap C_2)$。]

*41. 证明一个集合上的所有划分的集合与关系$\preccurlyeq$构成一个格，其中如果划分 P_1 是划分 P_2 的加细，则 $P_1\preccurlyeq P_2$。(见节 6.5 的练习 27 前面的说明。)

42. 证明每个全序集都是一个格。

43. 证明每个有限格都有一个最小元素和一个最大元素。

44. 给出一个无限格的例子，使得

   a) 既没有一个最小元素也没有一个最大元素。

   b) 有一个最小元素但没有一个最大元素。

   c) 有一个最大元素但没有一个最小元素。

   d) 有一个最小元素也有一个最大元素。

45. 验证$(\mathbf{Z}^+\times\mathbf{Z}^+,\preccurlyeq)$是一个良序集，其中$\preccurlyeq$是如例 7 所声明的字典顺序。

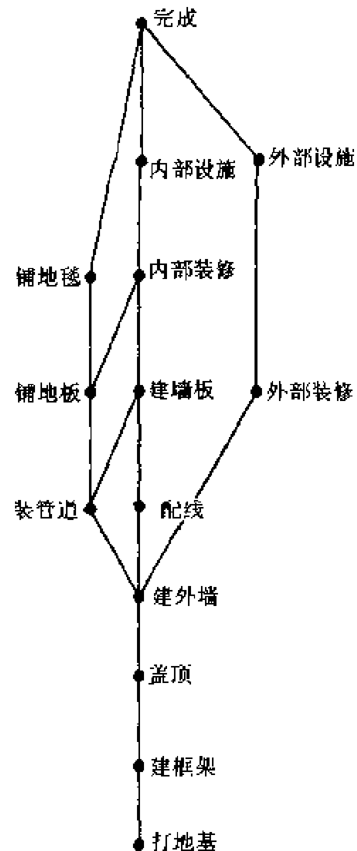
46. 证明一个有穷非空偏序集有一个极大元素。

47. 求一个全序使得它与练习 24 的哈斯图所表示的偏序集相容。

48. 求一个与集合$\{1,2,3,6,8,12,24,36\}$上的整除关系相容的全序。

49. 求一个为完成开发项目中任务的全序，使得它与例 26 构造的序不同。

50. 如果表示建筑一座房子所需任务的哈斯图如右图所示，通过指定这些任务的顺序来安排它们。



关键术语和结果

术语

从 A 到 B 的二元关系：$A\times B$ 的子集

A 上的关系：从 A 到自身的二元关系（即 $A\times A$ 的子集）

$S\circ R$：R 与 S 的合成

R^{-1}：R 的逆关系

R^n：R 的 n 次幂

自反的：A 上的一个关系是自反的，如果对所有的 $a\in A$ 有 $(a,a)\in R$

对称的：A 上的一个关系是对称的，如果只要 $(a,b)\in R$ 就有 $(b,a)\in R$

反对称的：A 上的关系是反对称的，如果只要 $(a,b)\in R$ 和 $(b,a)\in R$ 就有 $a=b$

传递的：A 上的关系 R 是传递的，如果从 $(a,b)\in R$ 和 $(b,c)\in R$ 推出 $(a,c)\in R$

A_1，A_2，…，A_n 上的 n 元关系：$A_1\times A_2\times\cdots\times A_n$ 的子集

关系数据模型：一个使用 n 元关系表示数据库的模型

主键码：n 元关系的一个域，使得一个 n 元组被它在这个域的值唯一确定

复合键码：一个 n 元关系域的笛卡儿积，使得一个 n 元组被它在这些域的值唯一确定

投影：一个函数，它从一个 n 元关系通过删除域产生一个阶较小的关系

连接：一个函数，它把具有某些相同的域的 n 元关系组合起来

有向图：称为顶点的元素以及这些元素的有序对（也叫作边）的集合

环：形如 (a,a) 的边

关系 R 关于性质 **P** 的闭包：包含 R 的关系 S（如果存在的话）具有性质 **P**，并且被任何包含 R 且具有性质 **P** 的关系所包含

有向图中的路径：边的序列 $(a,x_1),(x_1,x_2),\cdots,(x_{n-2},x_{n-1}),(x_{n-1},b)$ 使得序列中每条边的终点是后面一条边的始点

有向图的回路（或圈）：以同一顶点作为始点和终点的路径

R^*（连通关系）：由有序对 (a,b) 构成的关系，条件是存在一条从 a 到 b 的路径

等价关系：自反的、对称的和传递的关系

等价：如果 R 是等价关系，若 aRb，那么 a 等价于 b

$[a]_R$（a 关于 R 的等价类）：A 中所有等价于 a 的元素的集合

$[a]_m$（模 m 的同余类）：与 a 模 m 同余的整数的集合

集合 S 的划分：一族两两不相交的非空子集且这些子集的并就是 S

偏序：自反的、反对称的和传递的关系

偏序集 (S,R)：集合 S 与这个集合上的偏序 R

可比的：偏序集 $(A,\preccurlyeq)$ 的元素 a 和 b 是可比的，如果 $a\preccurlyeq b$ 或 $b\preccurlyeq a$

不可比的：一个偏序集的元素不是可比的

全序（或线序）：一个偏序，并且它的每对元素都是可比的

全序（或线序）集：具有一个全序（或线序）的偏序集

字典顺序：笛卡儿积或串上的一个偏序（见 6.6.2 节）

哈斯图：偏序集的图表示，其中所有的环和由传递性导出的边不出现，并且顶点的位置指示

了边的方向

极大元素：偏序集的一个元素，它不小于这个偏序集的任何其他元素

极小元素：偏序集的一个元素，它不大于这个偏序集的任何其他元素

最小元素：偏序集的一个元素，它小于这个集合的所有其他元素

最大元素：偏序集的一个元素，它大于这个集合的所有其他元素

集合的上界：偏序集的一个元素，它大于这个集合的所有其他元素

集合的下界：偏序集的一个元素，它小于这个集合的所有其他元素

集合的最小上界：集合的一个上界，它小于所有其他的上界

集合的最大下界：集合的一个下界，它大于所有其他的下界

格：一个偏序集：其中每对元素都有一个最大下界和一个最小上界

良序集：一个偏序集$(S,\preccurlyeq)$其中$\preccurlyeq$是全序，并且 S 的每个非空子集都有最小元素

与一个偏序相容的全序：包含了给定偏序的一个全序

拓扑排序：用给定偏序构造一个相容的全序

结果

集合 A 上的关系 R 的自反闭包等于$R\cup\Delta$，其中 $\Delta=\{(a,a)|a\in A\}$。

集合 A 上的关系 R 的对称闭包等于 $R\cup R^{-1}$，其中 $R^{-1}=\{(b,a)|(a,b)\in R\}$。

一个关系的传递闭包等于从这个关系构成的连通关系。

求一个关系的传递闭包的沃舍尔算法（见 6.4.5 节）。

设 R 是等价关系，那么下面三个语句是等价的：(1) $a R b$；(2) $[a]_R\cap[b]_R\neq\varnothing$；(3) $[a]_R=[b]_R$。

集合 A 上的等价关系的等价类构成 A 的划分。相反，从一个划分可以构造一个等价关系使得等价类就是划分中的子集。

拓扑排序算法（见 6.6.6 节）。

复习题

1. a) 什么是集合上的关系？
   b) 一个 n 元素集合上有多少个关系？
2. a) 什么是自反关系？
   b) 什么是对称关系？
   c) 什么是反对称关系？
   d) 什么是传递关系？
3. 给出集合$\{1,2,3,4\}$上的关系的一个例子，使得它是
   a) 自反的、对称的但不是传递的。
   b) 不是自反的，是对称的和传递的。
   c) 自反的、反对称的，但不是传递的。
   d) 自反的、对称的和传递的。
   e) 自反的、反对称的和传递的。
4. a) 在一个 n 元素集合上有多少个自反的关系？
   b) 在一个 n 元素集合上有多少个对称的关系？

c) 在一个 n 元素集合上有多少个反对称的关系？

5. a) 解释在一个大学里怎样用一个 n 元关系表示有关学生的信息。
   b) 怎样用一个包含学生姓名、地址、电话号码、专业和平均成绩的5元关系构造包含学生姓名、专业和平均成绩的3元关系
   c) 怎样用包含学生姓名、地址、电话号码和专业的4元关系和包含学生姓名、学号、专业和学分数组合成一个单一的 n 元关系？
6. a) 解释怎样使用一个0-1矩阵表示有穷集上的关系。
   b) 解释怎样使用表示关系的0-1矩阵来确定这个关系是否为自反的、对称的和反对称的。
7. a) 解释怎样使用一个有向图来表示有穷集上的关系。
   b) 解释怎样使用表示关系的有向图来确定这个关系是否为自反的、对称的和反对称的。
8. a) 定义一个关系的自反闭包和对称闭包。
   b) 怎样可以构造一个关系的自反闭包？
   c) 怎样可以构造一个关系的对称闭包？
   d) 求集合{1,2,3,4}上的关系{(1,2),(2,3),(2,4),(3,1)}的自反闭包和对称闭包。
9. a) 定义一个关系的传递闭包。
   b) 一个关系的传递闭包能够通过包含所有的对 (a,c) 得到吗？其中 (a,c) 满足 (a,b) 和 (b,c) 属于 R。
   c) 描述求关系的传递闭包的两个算法。
   d) 求关系{(1,1),(1,3),(2,1),(2,3),(2,4),(3,2),(3,4),(4,1)}的传递闭包。
10. a) 定义等价关系。
    b) 集合 $\{a,b,c,d\}$ 上的哪些关系是包含了 (a,b) 和 (b,d) 的等价关系？
11. a) 证明模 m 同余关系是等价关系，其中 m 是正整数。
    b) 证明关系 $\{(a,b) \mid a \equiv \pm b \pmod 7\}$ 是整数集上的等价关系。
12. a) 什么是一个等价关系的等价类？
    b) 什么是模5同余关系的等价类？
    c) 什么是问题11 b) 中等价关系的等价类？
13. 解释在集合的等价关系与集合划分之间的联系。
14. a) 定义偏序。
    b) 证明正整数集合上的整除关系是偏序。
15. 解释怎样用集合 A_1 和 A_2 上的偏序定义在集合 $A_1 \times A_2$ 上的偏序。
16. a) 解释怎样构造有穷集上的偏序的哈斯图。
    b) 画出集合{2，3，5，9，12，15，18}上的整除关系的哈斯图。
17. a) 定义一个偏序集的极大元素和最大元素。
    b) 给出一个有3个极大元素的偏序集的例子。
    c) 给出一个有1个最大元素的偏序集的例子。
18. a) 定义格。
    b) 给出一个5元素偏序集是格的例子和一个5元素偏序集不是格的例子。
19. a) 证明一个格的每个有穷子集有一个最大下界和一个最小上界。

b）证明每个具有有限元素的格有一个最小元素和一个最大元素。

20. a）定义一个良序集。

b）描述一个从偏序集产生良序集的算法。

c）如果每个任务仅当某个或某些其他任务完成以后可以开始，解释怎样用 b）中的算法排序这个项目中的任务。

补充练习

1. 设 S 是所有英语字母串的集合。确定下面的关系是否是自反的、反自反的、对称的、反对称的和传递的。
   a）$R_1=\{(a,b)\mid a$ 和 b 没有公共字母$\}$。
   b）$R_2=\{(a,b)\mid a$ 和 b 长度不相等$\}$。
   c）$R_3=\{(a,b)\mid a$ 比 b 长$\}$。
2. 构造集合 $\{a,b,c,d\}$ 上的关系，使得它是
   a）自反的和对称的，但不是传递的。
   b）反自反的、对称的和传递的。
   c）反自反的和反对称的，但不是传递的。
   d）自反的，既不是对称的也不是反对称的，是传递的。
   e）既不是自反的、反自反的、对称的和反对称的，也不是传递的。
3. $\mathbf{Z}\times\mathbf{Z}$ 上的关系 R 定义如下：$(a,b)R(c,b)$，当且仅当 $a+b=b+c$。证明 R 是等价关系。
4. 证明一个反对称关系的子集也是一个反对称关系。
5. 设 R 是 A 上的自反关系。证明 $R\subseteq R^2$。
6. 假设 R_1 和 R_2 是集合 A 上的自反关系。证明 $R_1\oplus R_2$ 是反自反的。
7. 假设 R_1 和 R_2 是集合 A 上的自反关系。$R_1\cap R_2$ 也是自反的吗？$R_1\cup R_2$ 也是自反的吗？
8. 假设 R 是集合 A 上的对称关系。$\overline{R}$ 也是对称的吗？
9. 设 R_1 和 R_2 是集合 A 上的对称关系。$R_1\cap R_2$ 也是对称的吗？$R_1\cup R_2$ 也是对称的吗？
10. 一个关系 R 叫做循环的，如果从 $a\,R\,b$ 和 $b\,R\,c$ 推出 $c\,R\,a$。证明 R 是自反的和循环的，当且仅当它是等价关系。
11. 证明一个 n 元关系的主键码也是这个关系的任何投影的主键码，其中这个投影包含这个键码作为它的一个域。
12. 一个 n 元关系的主键码也是由取这个关系与第二个关系的连接而得到的较大的关系的主键码吗？
13. 证明一个关系的对称闭包的自反闭包和它的自反闭包的对称闭包是相同的。
14. 设 R 是所有数学家的集合上的关系，R 包含有序对 (a,b)，当且仅当 a 与 b 合写了一篇论文，
    a）描述关系 R^2。
    b）描述关系 R^*。


c）如果一个数学家与多产的匈牙利数学家保罗·厄多斯[⊖]合写了一篇论文，那么这个数学家的厄多斯数是1。如果这个数学家没有与厄多斯合写一篇论文，但是与某个与厄多斯合写过论文的人合写了一篇论文，那么这个数学家的厄多斯数是2，依比下去（除了厄多斯本人的厄多斯数是0）。用 R 中的路径给出厄多斯数的定义。

*15．a）给出一个例子，证明一个关系的对称闭包的传递闭包不一定与这个关系的传递闭包的对称闭包相等。

b）证明一个关系的对称闭包的传递闭包一定包含这个关系的传递闭包的对称闭包。

16．a）设 S 是一个计算机问题的子程序的集合。定义关系 R，如果在执行中子程序 **P** 调用子程序 **Q**，那么 **P** R **Q**。描述 R 的传递闭包。

b）对于哪些子程序 **P**，(**P**,**P**)属于 R 的传递闭包？

c）描述 R 的传递闭包的自反闭包。

17．假设 R 和 S 是集合 A 上的关系，$R \subseteq S$，且 R 和 S 的关于性质 **P** 的闭包存在。证明 R 关于 **P** 的闭包是 S 关于 **P** 的闭包的子集。

18．证明两个关系的并的对称闭包是它们的对称闭包的并。

*19．设计一个基于内点概念的算法，求有向图中两个顶点之间的最长路径的长度，或确定在这些顶点之间存在任意长的路径。

20．下面的哪些关系是所有人集合上的等价关系？

a）$\{(x,y) \mid x$ 与 y 有同样的星座$\}$

b）$\{(x,y) \mid x$ 与 y 出生在同一年$\}$

c）$\{(x,y) \mid x$ 与 y 曾去过同一城市$\}$

*21．在5元素集合上有多少个不同的等价关系恰有3个不同的等价类？

22．证明 $\{(x,y) \mid x-y \in \mathbf{Q}\}$ 是实数集上的等价关系，其中 $\mathbf{Q}$ 是有理数集合。$[1]$，$[1/2]$，$[\pi]$ 是什么？


⊖ 保罗·厄多斯（Paul Erdös，1913—1996） 厄多斯生于匈牙利的布达佩斯，是两个高中数学教师的儿子。他是一个神童：3岁就能够心算3位数的乘法，4岁发现了负数。由于他的妈妈不想让他外出受到传染病的感染，他主要是在家读书。在17岁厄多斯进入Eötvös大学，4年毕业获得数学博士。毕业后作博士后，在英国曼彻斯特度过4年。在1938年由于匈牙利困难的政治形势，特别是对犹太人的形势，他去了美国。他在美国度过了一生的大多数时间，除了1954—1962年之外，这段时期由于麦卡锡偏执狂，他被禁止留在美国。他也在以色列度过相当长的时间。

厄多斯对组合数学和数论作出许多突出的贡献。他最值得骄傲的发现之一是关于素数定理的基础证明（这里的含义是指没有使用任何复杂的分析），这个定理对于不超过一个固定正整数的素数的个数进行了估计。在拉姆赛理论的近代发展他也起了作用。

厄多斯周游世界，出席会议，访问大学和研究所，与其他的数学家一起工作。他全身心地把自己献身于数学，从一个数学家到下一个数学家地旅行，宣称"我的大脑是开放的"。厄多斯是大约1500篇论文的作者或合作者，他几乎有500个合作者。因为他没有固定的家，这些论文的备份由AT&T实验室的一位著名的离散数学家劳恩·格雷汉姆保存，这个数学家与他广泛合作并给予他许多生活上的照顾。

厄多斯提供奖金，从10美元到10 000美元，用于他特别感兴趣的问题的求解，奖金的多少依赖于问题的难度。他付出过将近4 000美元。厄多斯有他自己的特殊语言，使用如"epsilon"（孩子），"boss（老板）"（女人），"slave（奴隶）"（男人），"captured（俘获）"（结婚），"liberated（自由）"（离婚），"Supreme Fascist（极端法西斯）"（上帝），"Sam"（美国）和"Joe"（苏联）等术语。尽管他在许多事上与众不同，但是他集中了几乎所有的能量在数学研究上。他没有嗜好，也没有全日制的职业。他从未结婚，看起来是独身主义者。厄多斯特别慷慨，他把来自奖品、奖金和薪金的许多钱用于学术交流和有价值的事情。他旅行特别简单，不像是有许多财产的人。

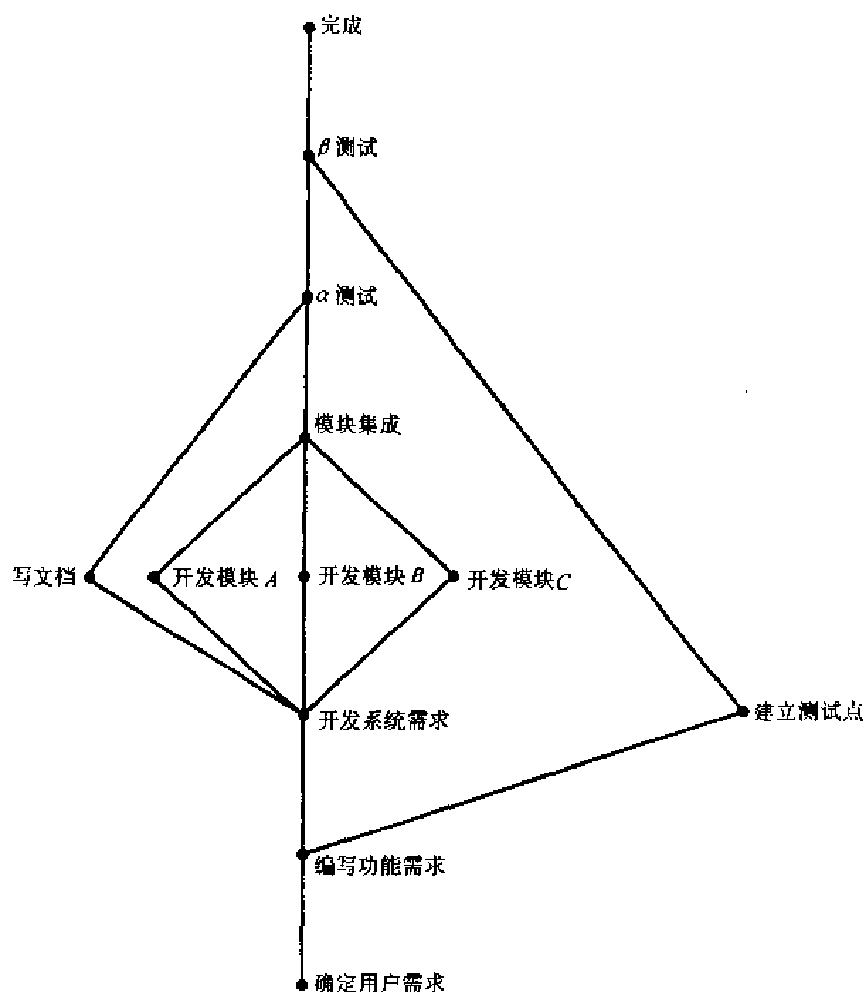
23. 设 $P_1=\{A_1,A_2,\cdots,A_m\}$ 和 $P_2=\{B_1,B_2,\cdots,B_n\}$ 都是集合 S 的划分。证明形如 $A_i\cap B_j$ 的非空子集族是 S 的划分，且是 P_1 和 P_2 的加细（见 6.5 节练习 27 前面的说明）。

*24. 证明关系 R 的自反闭包的对称闭包的传递闭包，是包含 R 的最小的等价关系。

25. 设 $\mathbf{R}(S)$ 是集合 S 上的所有关系的集合。如下定义 $\mathbf{R}(S)$ 上的关系 $\preccurlyeq$：如果 $R_1\subseteq R_2$ 则 $R_1\preccurlyeq R_2$，这里的 R_1 和 R_2 是 S 上的关系。证明 $(\mathbf{R}(S),\preccurlyeq)$ 是偏序集。

26. 设 $\mathbf{P}(S)$ 是 S 上的所有划分的集合。定义 $\mathbf{P}(S)$ 上的关系如下，如果 P_1 是 P_2 的加细，则 $P_1\preccurlyeq P_2$（见 6.5 节的练习 27）。证明 $(\mathbf{P}(S),\preccurlyeq)$ 是偏序集。

27. 对一个软件项目的任务进行排序，关于这个项目任务的哈斯图给在下面。



如果一个偏序集子集的每一对元素都是可比的，就称这个子集为一条链。如果它的每一对元素都是不可比的，就称这个子集为一条反链。

28. 找出 6.6 节练习 17～19 的哈斯图所表示的偏序集中的所有链。
29. 找出 6.6 节练习 17～19 的哈斯图所表示的偏序集中的所有反链。
30. 找出 6.6 节练习 24 的哈斯图所示的偏序集中具有最多元素的一条反链。
31. 证明在一个有穷偏序集 $(S,\preccurlyeq)$ 中每个极大链包含 S 的一个极小元素。（一条极大链是一条链，但不是一条更大的链的子链。）

** 32. 证明一个偏序集可以被分成 k 条链，其中 k 是这个偏序集中一条反链的最大元素数。

*33. 证明在任意一组 $mn+1$ 个人中或者存在 $m+1$ 个人一张表，其中每个人（除了表的第一个人以外）都是表中前一个人的后代，或者存在 $n+1$ 个人，其中没有一个人是其他 n 个人中任何一个人的后代。[提示：用练习 32。]

*34. 建立广义归纳原理：如果 $P(x_0)$ 为真，其中 x_0 是一个良序集 S 的最小元素（归纳基础），并且由 $P(x)$ 对于一切 $x \prec y$ 为真推出 $P(y)$ 为真（归纳步骤），那么对于 S 的每个元素 x，$P(x)$ 为真。

35. 使用在良序集 $(\mathbf{Z}^+ \cup \{0\} \times \mathbf{Z}^+ \cup \{0\})$ 上（关于字典顺序）的广义归纳原理证明 $a_{m,n} = [n(n+1)/2] + m$，其中 $a_{0,0} = 0$，并且

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1, & \text{如果 } n = 0 \\ a_{m,n-1} + n, & \text{如果 } n \neq 0 \end{cases}$$

集合 A 上的关系 R 叫做 A 上的拟序，如果 R 是自反的和传递的。

36. 设 R 是从 $\mathbf{Z}^+$ 到 $\mathbf{Z}^+$ 的所有函数的集合上的关系，使得 (f, g) 属于 R，当且仅当 f 是 $O(g)$。证明 R 是拟序。

37. 设 R 是 A 上的拟序，证明 $R \cap R^{-1}$ 是等价关系。

38. 设 R 是拟序，设 S 是 $R \cap R^{-1}$ 的等价类的集合上的关系，C 和 D 是 $R \cap R^{-1}$ 的等价类，(C, D) 属于 S，当且仅当存在 C 的元素 c 和 D 的元素 d 使得 (c, d) 属于 R。证明 S 是偏序。

设 L 是一个格。由 $x \wedge y = \mathrm{glb}(x, y)$ 和 $x \vee y = \mathrm{lub}(x, y)$ 定义交（$\wedge$）和联合（$\vee$）运算。

39. 证明下面的性质对格 L 的一切元素 x，y，z 成立：

a) $x \wedge y = y \wedge x$，$x \vee y = y \vee x$（交换律）

b) $(x \wedge y) \wedge z = x \wedge (y \wedge z)$，$(x \vee y) \vee z = x \vee (y \vee z)$（结合律）

c) $x \wedge (x \vee y) = x$，$x \vee (x \wedge y) = x$（吸收律）

d) $x \wedge x = x$，$x \vee x = x$（幂等律）

40. 证明如果 x 和 y 是格的元素，那么 $x \vee y = y$ 当且仅当 $x \wedge y = x$。

一个格 L 是有界的，如果它有一个上界，记作 $\mathbf{1}$，即对所有的 $x \in L$ 有 $x \preccurlyeq 1$，并且有一个下界，记作 0，即对所有的 $x \in L$ 有 $0 \preccurlyeq x$。

41. 证明如果个 L 是具有上界 1 和下界 0 的有界格，那么对所有的元素 $x \in L$ 下面的性质保持：

a) $x \vee 1 = 1$　　b) $x \wedge 1 = x$　　c) $x \vee 0 = x$　　d) $x \wedge 0 = 0$

42. 证明每个有限格是有界的。

一个格叫做分配的，如果对 L 中所有的 x，y，z 有 $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ 和 $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$。

*43. 给出一个不是分配格的例子。

44. 证明格 $(P(S), \subseteq)$ 是分配格，其中 $P(S)$ 是有穷集 S 的幂集。

45. 格 $(\mathbf{Z}^+, |)$ 是分配格吗？

有界格 L 的元素 a 关于上界 1 和下界 0 的补元是元素 b 使得 $a \vee b = 1$ 和 $a \wedge b = 0$。如果一个格的每个元素都有补元，那么这个格叫做有补格。

46. 给出一个有限格的例子，其中至少 1 个元素有多于 1 个的补元，且至少 1 个元素没有补元。
47. 证明格 $(P(S), \subseteq)$ 是有补格，其中 $P(S)$ 是有穷集 S 的幂集。
*48. 证明如果 L 是有限分配格，那么 L 的元素至多有 1 个补元。

计算机题目

用下面的输入和输出写程序。

1. 给定表示有穷集上关系的矩阵，确定这个关系是否是自反的或反自反的。
2. 给定表示有穷集上关系的矩阵，确定这个关系是否是对称的或反对称的。
3. 给定表示有穷集上关系的矩阵，确定这个关系是否是传递的。
4. 给定正整数 n，显示一个 n 元素集合上所有的关系。
*5. 给定一个正整数 n，确定 n 元素集合上的传递关系的个数。
*6. 给定一个正整数 n，确定 n 元素集合上的等价关系的个数。
*7. 给定一个正整数 n，显示 n 个最小的正整数集合上的所有的等价关系。
8. 给定一个 n 元关系，当某些特定的域被删除以后求这个关系的投影。
9. 给定一个 m 元关系、一个 n 元关系和一个公共字段的集合，找出这些关系关于这个公共字段的连接。
10. 给定表示一个有穷集上关系的矩阵，求表示这个关系自反闭包的矩阵。
11. 给定表示一个有穷集上关系的矩阵，求表示这个关系对称闭包的矩阵。
12. 给定表示一个有穷集上关系的矩阵，通过计算表示这个关系的矩阵的幂的联合求表示这个关系传递闭包的矩阵。
13. 给定表示一个有穷集上关系的矩阵，使用沃舍尔算法求表示这个关系的传递闭包的矩阵。
14. 给定表示一个有穷集上关系的矩阵，求表示包含这个关系的最小的等价关系的矩阵。
15. 给定一个有穷集上的偏序，使用拓扑排序找出一个与它相容的全序。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 显示一个 4 元素集合上的所有不同的关系。
2. 显示一个 6 元素集合上的所有不同的自反的和对称的关系。
3. 显示一个 5 元素集合上的所有不同的自反的和传递的关系。
*4. 对所有的正整数 n，$n \leqslant 7$，确定在 n 元素集合上存在有多少个传递的关系。
5. 在至少 20 个元素的集合上求某个关系的传递闭包。可以使用对应于某个特定运输或通信网络的有向链路的关系，或者使用一个随机生成的关系。
6. 对于所有不超过 20 的正整数 n，计算 n 元素集合上不同的等价关系个数。
7. 显示 7 元素集合上的所有等价关系。

*8. 显示 5 元素集合上的所有偏序。

*9. 显示 5 元素集合上的所有格。

写作题目

用课本以外的资料写成短文回答下列问题。

1. 讨论模糊关系的概念。怎样使用模糊关系？
2. 越出 6.2 节所讲述的内容，描述关系数据库的基本原理。关系数据库与其他类型的数据库相比使用得有多广？
3. 查找沃舍尔和罗伊的原始论文(法文)，在那篇论文中他们提出了求传递闭包的算法。讨论他们的方法。为什么可以认为称作沃舍尔算法的方法是由多人独立发现的？
4. 描述怎样用等价类把有理数定义为整数对的类，并且遵照这种方法怎样定义有理数的基本算术运算（见 6.5 节练习 10)。
5. 解释赫尔姆·哈斯怎样使用现在称为哈斯图的图示。
6. 描述在计算机操作系统中用来执行信息流策略的某些机制。
7. 讨论计划评审技术(PERT)在安排一个大的复杂项目的任务中的应用。PERT 使用得有多广？
8. 讨论关键路径方法(CPM)对找出完成项目的最短时间的应用。CPM 使用得有多广？
9. 讨论格中的对偶性的概念。解释怎样用对偶性建立新的结果？
10. 解释模格的意义。描述模格的某些性质，描述模格是怎样在投影几何的研究中产生的。

第7章　图

图论是有许多现代应用的古老题目。伟大的瑞士数学家列昂哈德·欧拉在18世纪引进了图论的基本思想。他利用图解决了有名的哥尼斯堡七桥问题，在本章里将要讨论这个问题。

图可用来解决许多领域的问题。例如，用图来确定能否在平面电路板上实现电路。用图来区别分子式相同但结构不同的两种化合物。用计算机网络的图模型来确定两台计算机是否由通信链路所连接。用边上带权值的图来解决诸如寻找交通网络里两个城市之间最短通路的问题。还可用图来安排考试和分配电视台的频道。

7.1　图的介绍

图是由一些顶点和连接这些顶点的一些边所组成的离散结构。存在多种不同类型的图，其间的区别在于连接顶点对的边的种类和数目。几乎每一门可以想象的学科里的问题都可以用图模型来解决。将要给出例子来说明如何用图作为不同领域的模型。例如，将要说明如何用图来表示生态环境里不同物种的竞争，如何用图来表示在组织里谁影响谁，以及如何用图来表示比赛结果。稍后将要说明如何用图解决许多问题，比如计算航线网络里两个城市之间航班的不同组合的数目，确定是否可能走遍城市里所有街道而不重复经过街道，以及求地图区域着色所需要的颜色数。

7.1.1　图的种类

将要通过说明如何用每一类型的图为计算机网络建模来介绍不同类型的图。假设网络包括计算机和计算机之间的电话线。用点表示每台计算机的位置，用弧表示每条电话线，如图7-1所示。

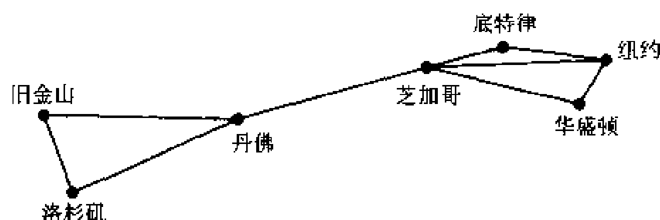


图7-1　计算机网络

对图7-1的网络有下述的观察结果。这个网络里两台计算机之间最多有一条电话线，每条线都是双向传输，没有计算机有到自身的电话线。所以这个网络可以用简单图建模，由表示计算机的顶点和表示电话线的无向边所组成，其中每条边都连接两个不同顶点，而且没有两条边连接同样一对顶点。

定义1　简单图$G=(V,E)$是由非空顶点集V和边集E所组成的，V的不同元素的无序对称为边。

有时网络里的计算机之间有多条电话线。当计算机之间通信量很大时情况就是这样的。图 7-2 显示带多重线的网络。简单图不能为这样的网络建模。取而代之的是利用多重图，它包括顶点和顶点之间的无向边，而且允许顶点对之间有多重边。每一个简单图也是多重图。不过，并非所有多重图都是简单图，因为在多重图里两条或更多条边可以连接同一对顶点。

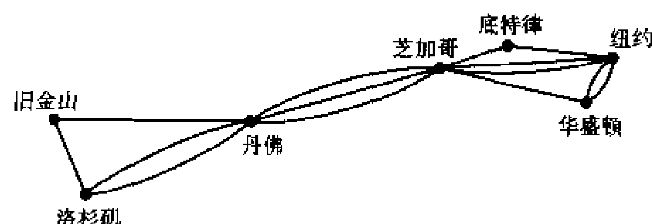


图 7-2　带多重线的计算机网络

当出现多重边时，不能用一对顶点指定图的边。这样使得多重图的形式化定义变得有点复杂。

定义 2　多重图 $G=(V,E)$ 是由非空顶点集 V、边集 E 以及从 E 到 $\{\{u,v\}\mid u,v\in V, u\neq v\}$ 的函数 f 所组成的。若 $f(e_1)=f(e_2)$，则边 e_1 和边 e_2 称为多重边或平行边。

计算机网络可以包含从计算机到它自身的电话线（可能是为了诊断的目的）。图 7-3 显示这样的网络。不能用多重图为这样的网络建模，因为多重图里不允许有环（它是从顶点到它自身的边）。取而代之的是利用伪图。伪图比多重图更为一般，因为在伪图里的边可以连接顶点和它自身。

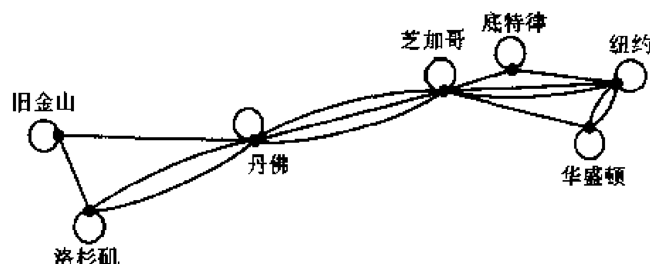


图 7-3　带诊断线的计算机网络

为了形式化地定义伪图，必须设法把边关联到只包含一个顶点的集合。

定义 3　伪图 $G=(V,E)$ 是由非空顶点集 V、边集 E 以及从 E 到 $\{\{u,v\}\mid u,v\in V\}$ 的函数 f 所组成的。若对某个 $u\in V$ 来说有 $f(e)=\{u,u\}=\{u\}$，则边 e 是环。

读者应当注意在伪图里多重边关联同一对顶点。不过，若至少有一条边 e 满足 $f(e)=\{u,v\}$，则说 $\{u,v\}$ 是图 $G=(V,E)$ 的边。我们不区分边 e 和它所关联的集合 $\{u,v\}$，除非识别个别多重边是重要的。

总而言之，伪图是最一般类型的无向图，因为它们包含环和多重边。多重图是包含多重边但没有环的无向图。最后，简单图是不带多重边和环的无向图。

计算机网络里电话线可以不是双向传输。例如，图 7-4 里纽约的主机只从其他计算机接

收数据而不发出数据。其他电话线都是双向传输，而且用成对的方向相反的边来表示。

用有向图（在第6章里研究它们）为这样的网络建模。有向图的边是有序对。允许有环（即相同元素的有序对），但不允许在两个顶点之间有同向的多重边。回忆下述的定义。

定义 4 有向图（V，E）是由非空顶点集 V、边集 E 所组成的，边是 V 中元素的有序对。

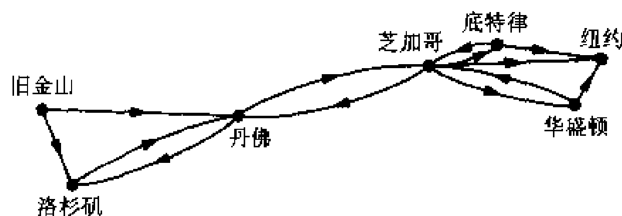


图 7-4 带单向电话线的通信网络

最后，计算机网络里可以出现多重线，所以从每个地点到纽约主机有多条单向线，并且从主机回到每个远程计算机有不止一条线。图 7-5 显示这样的网络。有向图不足以为这样的网络建模，因为这些图里不允许多重边。取而代之的是需要*有向多重图*，它有从一个顶点到第二个（可能相同）顶点的有向多重边。有向多重图的形式化定义如下。

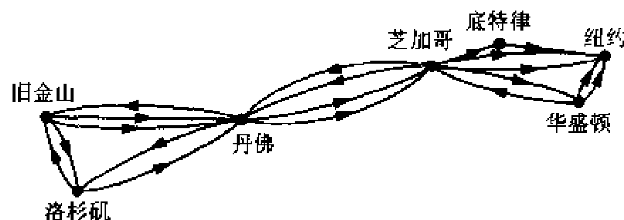


图 7-5 带多重单向线的计算机网络

定义 5 有向多重图$G=(V,E)$是由非空顶点集 V、边集 E 以及从 E 到$\{(u,v)|u,v\in V\}$的函数 f 所组成的。若$f(e_1)=f(e_2)$，则边 e_1 和边 e_2 是多重边。

读者应当注意到，多重有向边所关联的是同一对顶点。不过，只要至少有一条边满足$f(e)=(u,v)$，就说(u,v)是$G=(V,E)$的边。不区分边 e 和它所关联的有序对（u，v），除非识别个别多重边是重要的。

对各种类型的图来说，这样的术语表达清楚了图的边是关联着无序对还是有序对，是否允许多重边，以及是否允许环。将用图来描述带有有向边或无向边、带有或不带环和多重边的图。将用名词*无向图*或*伪图*来表示带有多重边和环的无向图。当说到边关联着有序对的图时，，将总是利用形容词*有向的*。表 7-1 总结了各种类型的图的定义。因为对图论的相对现代的兴趣，以及图论对各式各样的学科的应用，有许多不同的图论术语都是常用的。每当遇到这样的名词时，读者应当自行决定它们是如何使用的。也许有朝一日这样的术语将变得标准化。

表 7-1 图的术语

类 型	边	允许多重边	允许环
简单图	无向	否	否
多重图	无向	是	否
伪图	无向	是	是
有向图	有向	否	是
有向多重图	有向	是	是

7.1.2 图模型

图可用在各种模型里。在这里将要给出不同领域的图模型。本章和后面几章的后续小节将要介绍其他模型。

例 1 *生态学里栖息地重叠图* 图可用在涉及到不同种类的动物在一起活动的许多模型里。例如用*栖息地重叠图*为生态系统里物种之间的竞争建模。用顶点表示每个物种。若两个物种竞争（即它们共享某些食物来源），则用无向边连接代表它们的顶点。图 7-6 里的图表示森林生态系统。从这个图看出松鼠与浣熊竞争，但是乌鸦不与鼩鼱竞争。■

例 2 *影响图* 在对群体行为的研究里，可以观察到某些人能够影响其他人的思维。一种称为*影响图*的有向图可以用来为这样的行为建模。用顶点表示群体的每个人。当顶点 a 所表示的人影响到顶点 b 所表示的人时，就有从顶点 a 到顶点 b 的有向边。图 7-7 是表示群体成员的影响图的例子。在这个用影响图建模的群体里，Deborah 影响 Brian、Fred 以及 Linda，但是没有人影响 Deborah。另外，Yvonne 与 Brian 互相影响。■

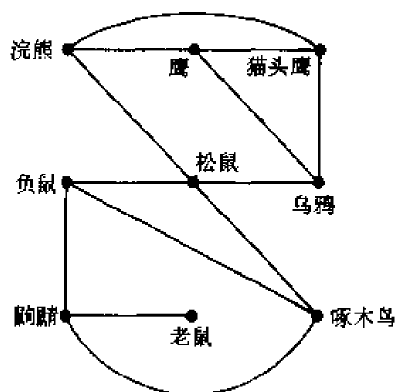


图 7-6 栖息地重叠图

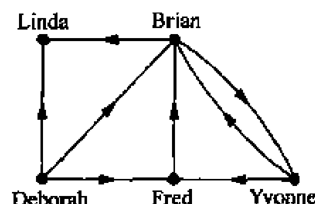


图 7-7 影响图

例 3 *巡回联赛* 每个队都与其他每队恰好比赛一次的联赛称为*巡回联赛*。用其中顶点表示每个队的有向图来为这样的联赛建模。注意若 a 队击败 b 队，则 (a, b) 是边。图 7-8 给出这样的有向图。注意在这个联赛里，队 1 无败绩而队 3 无胜绩。■

例 4 *优先图与并发处理* 通过并发地执行某些语句，计算机程序可以执行得更快。重要的是避免执行需要尚未执行语句结果的语句。语句与前面语句的相关性可以表示成有向

图。用顶点表示每个语句，若在执行完第一个顶点所表示的语句之前不能执行第二个顶点所表示的语句，则从第一个顶点到第二个顶点有一条边。这样的图称为优先图。图 7-9 显示计算机程序和优先图。例如，该图说明在执行语句 S_1、S_2 和 S_4 之前不能执行语句 S_5。 ■

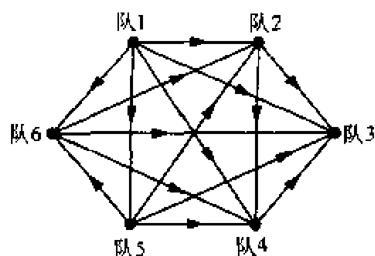


图 7-8　巡回联赛的图模型

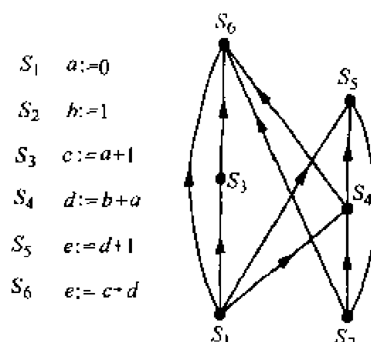
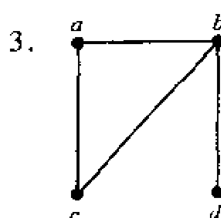


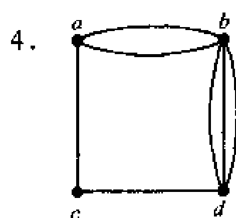
图 7-9　优先图

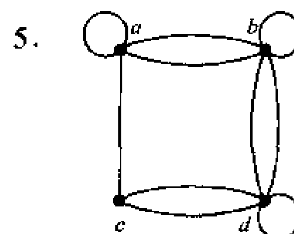
练习

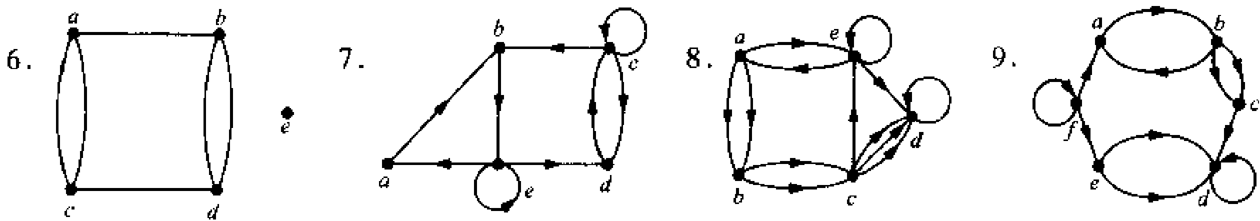
1. 画出表示航空公司航班路线的图模型，并说出所用图的类型，其中每天有 4 个航班从波士顿到纽瓦克，2 个航班从纽瓦克到波士顿，3 个航班从纽瓦克到迈阿密，2 个航班从迈阿密到纽瓦克，1 个航班从纽瓦克到底特律，2 个航班从底特律到纽瓦克，3 个航班从纽瓦克到华盛顿，2 个航班从华盛顿到纽瓦克，1 个航班从华盛顿到迈阿密，其中
   a）若城市之间有航班（任何方向），则在表示城市的顶点之间有边。
   b）对城市之间每个航班（任何方向）来说，在表示城市的顶点之间有边。
   c）对城市之间每个航班（任何方向）来说，在表示城市的顶点之间有边，此外对在迈阿密起飞和降落的特殊观光旅行增加一个环。
   d）从表示航班出发城市的顶点到表示航班终止城市的顶点之间有边。
   e）对每个航班从表示出发城市的顶点到表示终止城市的顶点之间有边。
2. 什么类型的图可以用来为大城市之间的高速公路系统建模？其中
   a）若在城市之间有州际高速公路，则在表示城市的顶点之间有边。
   b）对城市之间每条州际高速公路，在表示城市的顶点之间有边。
   c）对城市之间每条州际高速公路，在表示城市的顶点之间有边，此外若有环城州际高速公路，则在表示该城的顶点上有环。

对练习 3～9，判断所示的图是否简单图、多重图（非简单图）、伪图（非多重图）、有向图或有向多重图（非有向图）。

3. 

4. 

5. 



10. 对练习 3~9 里每个非简单无向图，找出删除后使它变成简单图的边。
11. 集合 A_1，…，A_n 的交图对每个集合来说都有一个顶点，若两个集合有非空交集，则有一条边连接代表这两个集合的顶点。构造下列集合的交图。

a) $A_1=\{0, 2, 4, 6, 8\}$，$A_2=\{0, 1, 2, 3, 4\}$，

$A_3=\{1, 3, 5, 7, 9\}$，$A_4=\{5, 6, 7, 8, 9\}$，

$A_5=\{0, 1, 8, 9\}$

b) $A_1=\{\ldots, -4, -3, -2, -1, 0\}$，

$A_2=\{\ldots, -2, -1, 0, 1, 2, \ldots\}$，

$A_3=\{\ldots, -6, -4, -2, 0, 2, 4, 6, \ldots\}$

$A_4=\{\ldots, -5, -3, -1, 1, 3, 5, \ldots\}$，

$A_5=\{\ldots, -6, -3, 0, 3, 6, \ldots\}$

c) $A_1=\{x \mid x<0\}$，

$A_2=\{x \mid -1<x<0\}$，

$A_3=\{x \mid 0<x<1\}$，

$A_4=\{x \mid -1<x<1\}$，

$A_5=\{x \mid x>-1\}$，

$A_6=\mathbf{R}$

12. 用图 7-6 里的栖息地重叠图来确定与鹰竞争的物种。
13. 构造 6 种鸟的栖息地重叠图，其中隐士鸫与旅鸫并与蓝松鸦竞争，旅鸫也与嘲鸫竞争，嘲鸫也与蓝松鸦竞争，以及鸸鸟与多毛啄木鸟竞争。
14. 在例 2 的影响图里谁影响 Fred？ Fred 影响谁？
15. 构造公司董事会成员的影响图，总裁影响研发董事、市场董事和执行董事；研发董事影响执行董事；市场董事影响执行董事；无人影响首席财务官或受其影响。
16. 在图 7-8 表示的巡回联赛里，队 4 击败哪些其他队？哪些队击败队 4？
17. 在巡回联赛里，老虎队击败蓝松鸦队，老虎队击败红衣主教队，老虎队击败金莺队，蓝松鸦队击败红衣主教队，蓝松鸦队击败金莺队，红衣主教队击败金莺队，用有向图为这样的结果建模。

18. 在例 4 的程序里，执行 S_6 之前必须执行哪些语句？（用图 7-9 中的优先图。）
19. 构造下列程序的优先图：

$S_1: x := 0$

$S_2: x := x + 1$

$S_3: y := 2$

$S_4: z := y$

$S_5: x := x + 2$

$S_6: y := x + z$

$S_7: z := 4$

20. 描述一种用来为航空公司的航班路线和航班时间建模的基于图的离散结构。[提示：给一个有向图添加结构。]
21. 描述一种用来为群体里成对个人之间关系建模的基于图的离散结构，其中每个人可能对另一人喜欢或者不喜欢，或者中立，而反过来的关系可以是不同的。[提示：给一个有向图添加结构。分别处理表示两个人的顶点之间的反向边。]

7.2　图的术语

7.2.1　引言

在本节里将介绍图论的一些基本词汇。当解决许多不同类型的问题时就使用这样的词汇。其中一个这样的问题涉及到判定能否把图画在平面里，使得没有两条边是交叉的。另一个例子是判定两个图是否具有顶点之间的一一对应，使得这样的对应能够产生边之间的一一对应。还将介绍在例子和模型里经常用到的几族重要的图。

7.2.2　基本术语

首先给出描述无向图的顶点和边的一些术语。

定义 1　若 $\{u, v\}$ 是无向图 G 的边，则两个顶点 u 和 v 称为在 G 里邻接（或相邻）。若 $e = \{u, v\}$，则边 e 称为关联顶点 u 和 v。也可以说边 e 连接 u 和 v。顶点 u 和 v 称为边 e 的端点。

为了反映出有多少条边关联着一个顶点，有下述的定义。

定义 2　在无向图里，顶点的度是与该顶点关联的边的数目，例外的情形是，顶点上的环为顶点的度做出双倍贡献。顶点 v 的度表示成 $\deg(v)$。

例 1　图 7-10 所示图 G 和 H 的顶点的度是什么？

解　在 G 里，$\deg(a) = 2$，$\deg(b) = \deg(c) = \deg(f) = 4$，$\deg(d) = 1$，$\deg(e) = 3$，$\deg(g) = 0$。在 H 里，$\deg(a) = 4$，$\deg(b) = \deg(e) = 6$，$\deg(c) = 1$，$\deg(d) = 5$。■

把度为 0 的顶点称为孤立的。因此孤立点不与任何顶点相邻。例 1 里图 G 的顶点 g 是孤立的。顶点是悬挂的，当且仅当它的度是 1。因此悬挂点恰与 1 个其他顶点相邻。例 1 里图 G 的顶点 d 是悬挂的。

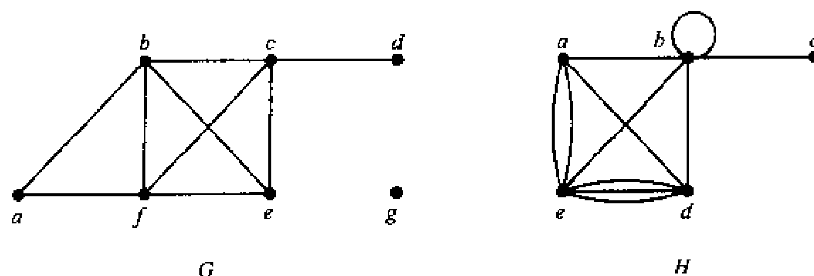


图 7-10 无向图 G 和 H

当对图$G=(V,E)$的所有顶点的度求和时，得出了什么？每条边都为顶点的度之和贡献2，因为一条边恰好关联2个（可能相同）顶点。这意味着顶点的度之和是边数的2倍。有下述的结果，它有时称为握手定理，这是因为在一条边有两个端点与一次握手涉及到两只手这两件事情之间的类比。

定理 1 *握手定理* 设$G=(V,E)$是有 e 条边的无向图，则

$$2e=\sum_{v\in V}\deg(v)$$

（注意即使出现多重边和环，这个式子也仍然是成立的。）

例 2 一个具有 10 个顶点而且每个顶点都度为 6 的图，有多少条边？

解 因为顶点的度之和是 $6\cdot10=60$，所以 $2e=60$。因此 $e=30$。■

定理 1 说明无向图顶点的度之和是偶数。这个简单事实有许多后果，其中一个后果作为定理 2 给出。

定理 2 无向图有偶数个奇数度顶点。

证 在无向图$G=(V,E)$里，设 V_1 和 V_2 分别是偶数度顶点和奇数度顶点的集合。于是

$$2e=\sum_{v\in V}\deg(v)=\sum_{v\in V_1}\deg(v)+\sum_{v\in V_2}\deg(v)$$

因为对 $v\in V_1$ 来说 $\deg(v)$ 是偶数，所以上面等式右端的第一项是偶数。另外，上面等式右端的两项之和是偶数，因为和是 $2e$。因此，和里的第二项也是偶数。因为在第二项的和里，所有的项都是奇数，所以必然有偶数个这样的项。因此，有偶数个奇数度顶点。■

对带有有向边的图来说也有一些有用的术语。

定义 3 当(u,v)是带有有向边的图 G 的边时，说 u 邻接到 v，而且说 v 从 u 邻接。顶点 u 称为(u,v)的*起点*，v 称为(u,v)的*终点*。环的起点和终点是相同的。

因为带有有向边的图的边是有序对，所以这时顶点度的定义细化成反映这个顶点作为起点和作为终点的不同度数。

定义 4 在带有有向边的图里，顶点 v 的*入度*（表示成$\deg^-(v)$）是以 v 作为终点的边数。顶点 v 的*出度*（表示成$\deg^+(v)$）是以 v 作为起点的边数。（注意顶点上的环对这个顶

点的入度和出度的贡献都是 1。)

例 3　求出图 7-11 所示带有向边的图 G 里每个顶点的入度和出度。

解　G 里的入度是：$\deg^-(a)=2$，$\deg^-(b)=2$，$\deg^-(c)=3$，$\deg^-(d)=2$，$\deg^-(e)=3$，$\deg^-(f)=0$。出度是：$\deg^+(a)=4$，$\deg^+(b)=1$，$\deg^+(c)=2$，$\deg^+(d)=2$，$\deg^+(e)=3$，$\deg^+(f)=0$。■

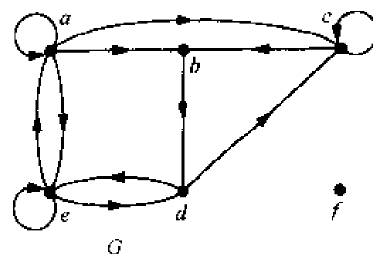


图 7-11　有向图 G

因为每条边都有一个起点和一个终点，所以在带有向边的图里，所有顶点的入度之和等于出度之和。这两个和都等于图的边数。把这个结果叙述成下述的定理。

定理 3　设 $G=(V,E)$ 是带有向边的图。于是

$$\sum_{v\in V}\deg^-(v)=\sum_{v\in V}\deg^+(v)=|E|$$

带有向边的图有许多性质是不依赖于边的方向的。因此忽略这些方向经常是有用处的。忽略边的方向后得出的无向图称为*底无向图*。带有向边的图与它的底无向图有相同的边数。

7.2.3　一些特殊的简单图

现在将要介绍几类简单图。常常用这些图作为例子，这些图来自许多应用里。

例 4　**完全图**　n 个顶点上的完全图（表示成 K_n）是在每对不同顶点之间都恰有一条边的简单图。对 $n=1, 2, 3, 4, 5, 6$ 来说，图 7-12 显示图 K_n。■

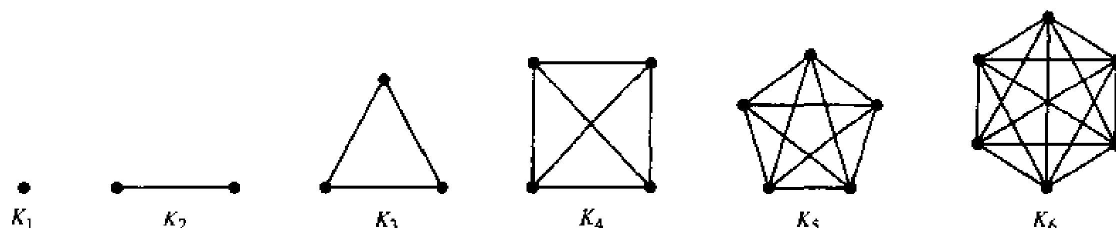


图 7-12　图 K_n，$n=1, 2, 3, 4, 5, 6$

例 5　**圈图**　圈图 $C_n(n\geqslant 3)$ 是由 n 个顶点 $v_1, v_2, \cdots, v_n$ 以及边 $\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ 组成的。图 7-13 显示圈图 C_3, C_4, C_5, C_6。■

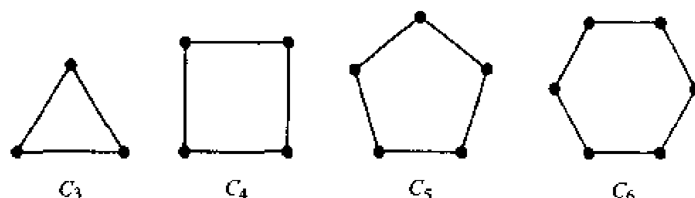


图 7-13　圈 C_3，C_4，C_5，C_6

例 6　**轮图**　对 $n\geqslant 3$ 来说，当给圈图 C_n 添加另一个顶点，而且把这个新顶点与 C_n 里 n 个顶点逐个连接时，就得出轮图 W_n。图 7-14 显示轮图 W_3, W_4, W_5, W_6。■

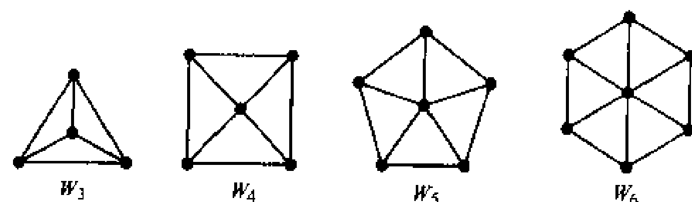


图 7-14 轮图 W_3，W_4，W_5，W_6

例 7 *n* 立方体图 *n* 立方体图（表示成 Q_n）是用顶点表示 2^n 个长度为 *n* 的位串的图。两个顶点相邻，当且仅当它们所表示的位串恰恰相差一位。图 7-15 显示图 Q_1，Q_2，Q_3。■

7.2.4 偶图

有时把图的顶点分成两个不相交的子集，使得每条边都连接一个子集里的顶点与另一个子集里的顶点。例如，考虑一下表示村民之间的婚姻的图，其中用顶点表示每个人，用边表示婚姻。在这个图里，每条边都连接表示男人的顶点子集里的顶点与表示女人的顶点子集里的顶点。这导致下述的定义。

定义 5 若把简单图 G 的顶点集合分成两个不相交的非空集合 V_1 和 V_2，使得图里的每一条边都连接着 V_1 里的一个顶点与 V_2 里的一个顶点（因此 G 里没有边是连接着 V_1 里的两个顶点或 V_2 里的两个顶点），则 G 称为偶图。

在例 8 里说明 C_6 是偶图，在例 9 里说明 K_3 不是偶图。

例 8 图 7-16 所示的 C_6 是偶图，因为把它的顶点集分成两个集合 $V_1=\{v_1, v_3, v_5\}$ 和 $V_2=\{v_2, v_4, v_6\}$，C_6 的每一条边都连接 V_1 里的一个顶点与 V_2 里的一个顶点。■

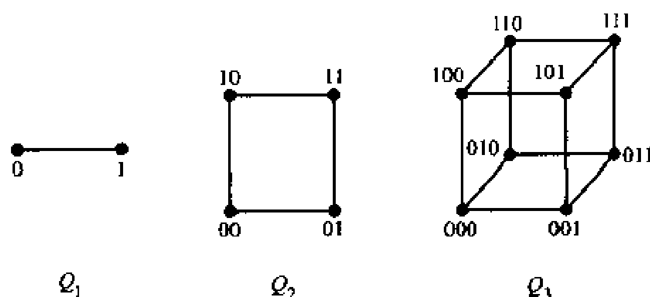


图 7-15 对于 $n=1, 2, 3$ 的 *n* 立方体图 Q_n

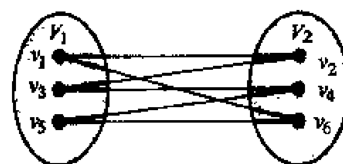


图 7-16 说明 C_6 是偶图

例 9 K_3 不是偶图。为了看明白这一点，注意若把 K_3 的顶点集合分成两个不相交的集合，则两个集合之一必然包含两个顶点。假如这个图是偶图，那么这两个顶点就不能用边连接，但是在 K_3 里每一个顶点都用边连接着到其他每个顶点。■

例 10 图 7-17 所示的图 G 和 H 是否为偶图？

解 图 G 是偶图，因为它的顶点集是两个不相交集合 $\{a, b, d\}$ 和 $\{c, e, d, f\}$ 的并，每条边都连接一个子集里的一个顶点与另一个子集里的一个顶点。（注意对 G 作为偶图来说，不必让 $\{a, b, d\}$ 里每一个顶点与 $\{c, e, d, f\}$ 里每一个顶点都相邻。例如 b 与 g 就不相邻。）

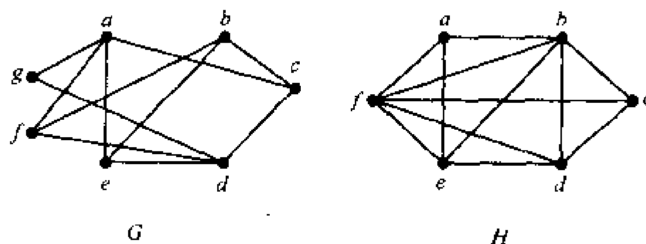


图 7-17 无向图 G 和 H

图 H 不是偶图，因为它的顶点集合不能分成两个子集，使得边都不连接同一个子集的两个顶点。（读者应当通过考虑顶点 a，b，f 来验证它。） ■

例 11 *完全偶图* 完全偶图 $K_{m,n}$是顶点集合分成分别含有 m 和 n 个顶点的两个子集的图。两个顶点之间有边，当且仅当一个顶点属于第一个子集而另一个顶点属于第二个子集。图 7-18 显示完全偶图 $K_{2,3}$，$K_{3,3}$，$K_{3,5}$，以及 $K_{2,6}$。 ■

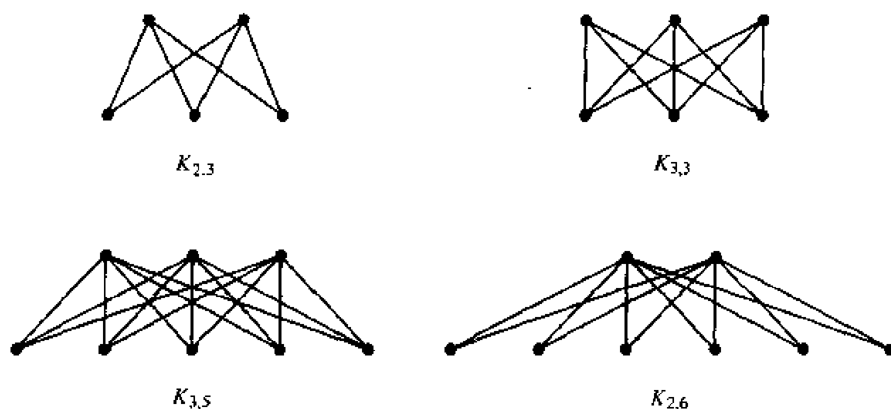


图 7-18 一些完全偶图

7.2.5 特殊类型的图的一些应用

将要说明如何把特殊类型的图用在数据通信和并行处理的模型里。

例 12 *局域网* 在一座大楼里，像小型计算机和个人电脑这样的各种计算机，以及像打印机和绘图仪这样的外设，都可以用局域网来连接。一些这样的网络是基于星型拓扑，其中所有设备都连接到中央控制设备。局域网用图 7-19 a）所示的完全偶图 $K_{1,n}$来表示。通过中央控制设备把消息从设备送到设备。

其他局域网是基于环型拓扑，其中每个设备恰恰连接到两个其他设备。带环型拓扑的局域网用图 7-19 b）所示的 n 圈图 C_n 来建模。围绕着圈把消息从设备送到设备，直到抵达消息目的地为止。

最后，一些局域网采用这两种拓扑的混合形式。消息是围绕着环或通过中央设备来传送。这样的冗余使得网络更加可靠。带冗余的局域网用图 7-19 c）所示的轮图 W_n 来建模。 ■

例 13 *并行计算的互联网络* 直到最近，计算机执行程序还是一次完成一个操作。因此，为解决问题而写的算法都设计成一次执行一步；这样的算法称为串行的。（几乎所有本

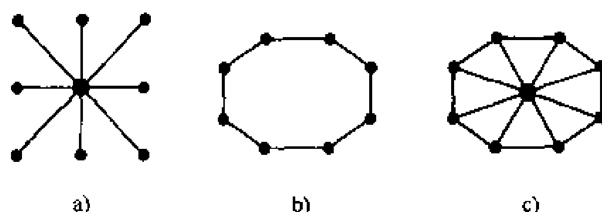


图 7-19 局域网的星型、环型以及混合拓扑

书描述的算法都是串行的。）不过，像气象模拟、医学图像以及密码分析这样的许多高强度计算问题，即使在超级计算机上，也不能通过串行操作在合理的时间范围之内解决。另外，对计算机执行基本操作有多快来说，还存在物理的限制，所以总是有问题不能用串行操作在合理长度的时间之内解决。

*并行处理*利用包含许多自带内存的单处理器的计算机来帮助克服串行计算机的局限性。*并行算法*把问题分成可并发解决的若干子问题，于是可以设计并行算法，用带有多处理器的计算机来快速解决问题。在并行算法里，单个指令流控制着算法的执行，包括把子问题送到不同的处理器，以及把子问题的输入和输出定向到适当的处理器。

采用并行处理时，一个处理器需要另一个处理器产生的输出。因此处理器需要互联。用适当类型的图来表示带有多重处理器的计算机里处理器的互联网络。在以下讨论里将要描述最常用类型的并行处理器互联网络。用来实现具体并行算法的互联网络的类型，依赖于处理器之间交换数据的需求量和所需要的速度，当然还有可用的硬件等。

最简单而且最昂贵的网络互联处理器，包含每对处理器之间的双向连接。当有 n 个处理器时，这样的网络表示成 n 个顶点上完全图 K_n。不过，这种类型的互联网络有严重的问题，因为需要的连接数太大。在现实里，处理器可以直接连接的数目是有限的，所以当处理器数目很大时，处理器不能直接连接到所有其他处理器。例如，当有 64 个处理器时，就需要 $C(64,2)=2016$ 个连接，每个处理器都得直接连接 63 个其他处理器。

另一方面，互联 n 个处理器的最简单方式或许是使用称为*线性阵列*的排列方式。除 P_1 和 P_n 外的每个处理器 P_i 都通过双向连接来连接到相邻处理器 P_{i-1} 和 P_{i+1}。P_1 只连接 P_2，P_n 只连接 P_{n-1}。图 7-20 显示 6 个处理器线性阵列。缺点是为了让处理器共享信息，有时需要使用大量的称为 **hop** 的中间连接。



图 7-20 6 个处理器的线性阵列

栅格网络（或*二维阵列*）是通用的互联网络。在这样的网络里，处理器个数是一个完全平方数，比方说 $n=m^2$。n 个处理器标记成 $P(i,j)$，$0\leqslant i\leqslant m-1$，$0\leqslant j\leqslant m-1$。双向连接把处理器 $P(i,j)$ 连接到它的 4 个相邻处理器 $P(i\pm1,j)$ 和 $P(i,j\pm1)$，只要这些处理器是在栅格里。（注意栅格角上的 4 个处理器只有 2 个相邻处理器，边界上其他处理器只有 3 个相邻处理器；有时也用每个处理器恰有 4 个连接的变种的栅格网络；见本节末尾的练习 44。）栅格网络限制每个处理器的连接数。某些成对处理器之间的通信需要 $O(\sqrt{n})=O(m)$ 个中间连接。（见本节末尾的练习 45。）图 7-21 显示表示 16 个处理器的栅格网络的图。

一个重要类型的互联网络是超立方体。对这样的网络来说，处理器数是 2 的幂，$n=2^m$。n 个处理器标记成 P_0，P_1，…，P_{n-1}。每个处理器都有到 m 个其他处理器的双向连

接。处理器 P_i 与下标的二进制表示与 i 的二进制表示恰恰相差 1 位的处理器相连接。超立方体网络在每个顶点的直接连接数与保证处理器通信的中间连接数之间取得了平衡。已经用超立方体网络建造了许多计算机，而且用超立方体网络设计了许多算法。n 立方体图 Q_n 表示带 n 个处理器的超立方体网络。图 7-22 显示 8 个处理器的超立方体网络。(图 7-22 显示出与图 7-15 所示的画 Q_3 的方式不同。) ■

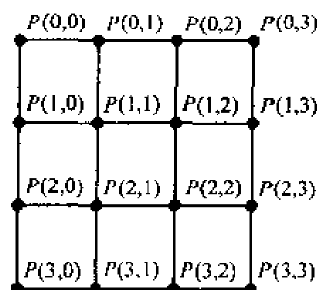


图 7-21 16 个处理器的栅格网络

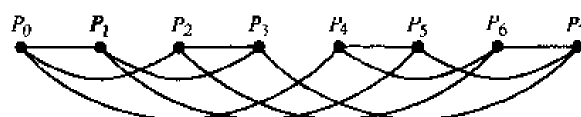


图 7-22 8 个处理器的超立方体网络

7.2.6 从旧图到新图

有时为了解决问题，只需要图的一部分。例如，只关心涉及在纽约、丹佛、底特律以及亚特兰大的计算机中心的大计算机网络的一部分。于是可以忽略其他的计算机中心以及不连接到这 4 个具体计算机中心里任何 2 个的所有电话线。在大网络的图模型里，可以删除与不是所关心的 4 个计算机中心所对应的顶点，可以删除与所删除顶点关联的所有边。当从图里删除一些边和顶点而不删除任何剩余的边的端点时，得出了较小的图。把这样的图称为原图的子图。

定义 6 图 $G=(V,E)$ 的子图是图 $H=(W,F)$，其中 $W\subseteq V$ 而且 $F\subseteq E$。

例 14 图 7-23 所示的图 G 是 K_5 的子图。 ■

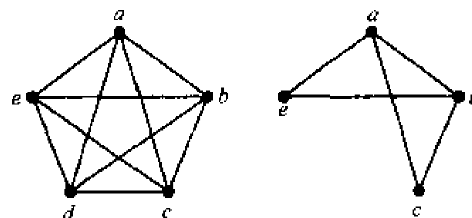


图 7-23 K_5 的子图

可以用各种方式组合两个或更多的图。把这些图的所有顶点和边所组成的图称为这些图的并图。将对 2 个简单图的并图给出更加形式化的定义。 ■

定义 7 两个简单图 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$ 的并图是带有顶点集 $V_1\cup V_2$ 和边集 $E_1\cup E_2$ 的简单图。G_1 和 G_2 的并图表示成 $G_1\cup G_2$。

例 15 求图 7-24 a) 所示的图 G_1 和 G_2 的并图。

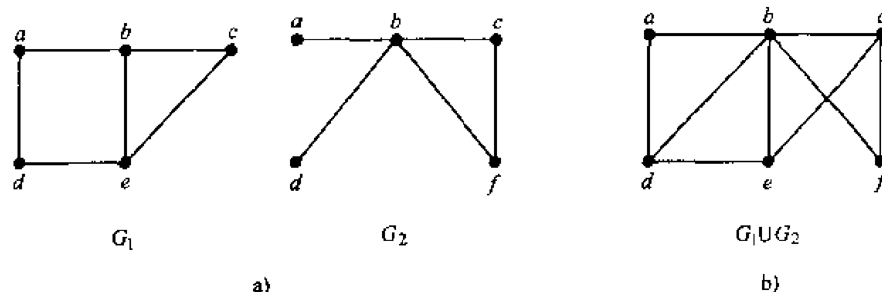
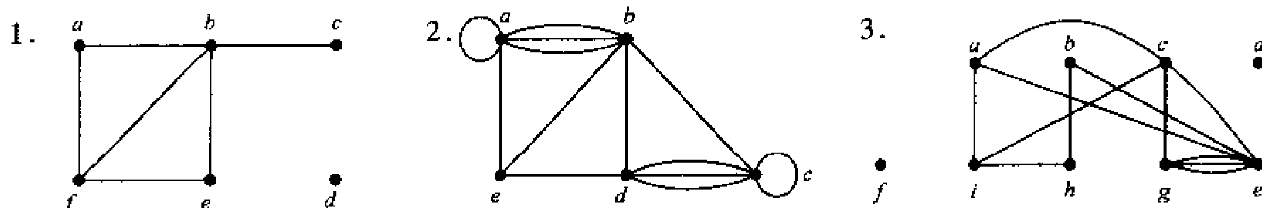


图 7-24 a) 简单图 G_1 和 G_2 以及 b) 它们的并 $G_1\cup G_2$

解　并图 $G_1 \cup G_2$ 的顶点集是两个顶点集的并，即 $\{a, b, c, d, e, f\}$。并图的边集是两个边集的并。并图显示图 7-24 b）中。■

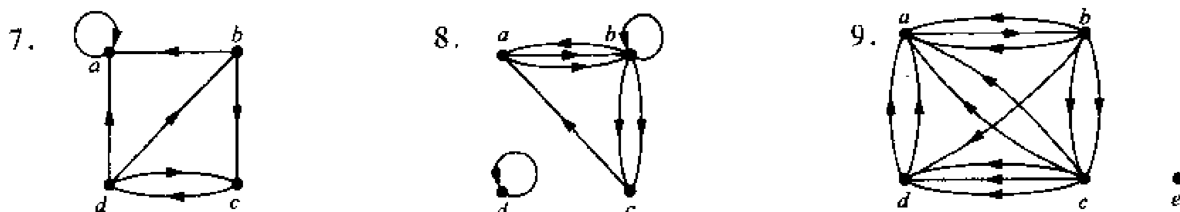
练习

在练习 1～3 里，求所给无向图的顶点数、边数以及每个顶点的度。指出所有孤立点和悬挂点。



4. 求练习 1～3 里每个图的顶点的度之和，并验证它等于图里边数的 2 倍。
5. 能否存在带有 15 个顶点而且每个顶点的度都为 15 的简单图？
6. 证明：在一次聚会上全体人员的握手次数之和是偶数。假设无人自己与自己握手。

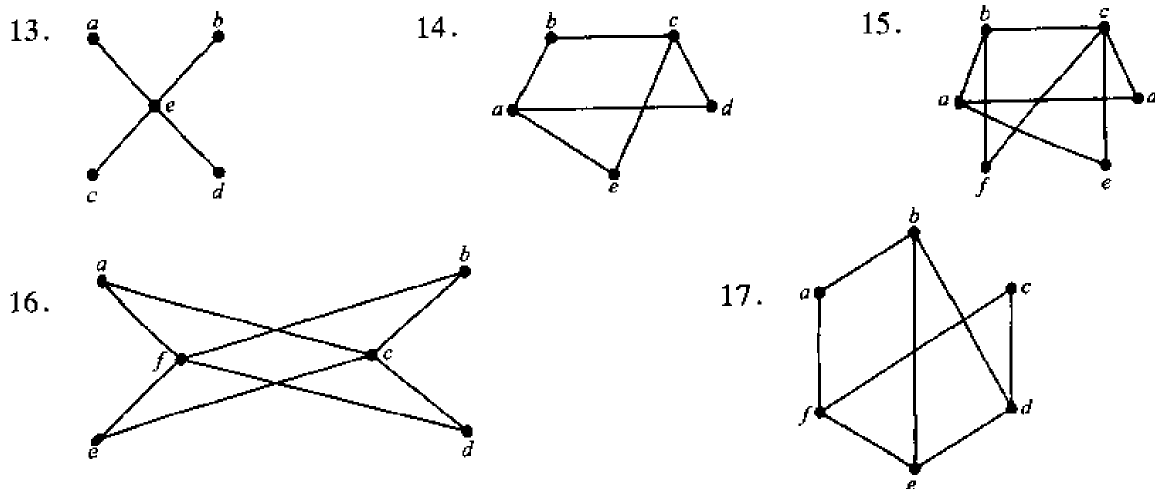
在练习 7～9 里，对给定的有向多重图来说，确定顶点数和边数，并求出每个顶点的入度和出度。



10. 对练习 7～9 里每个图来说，直接确定每个顶点的入度之和与出度之和。证明：它们都等于图里的边数。
11. 构造出图 7-11 里带有向边的图的底图。
12. 画出下列图。

    a) K_7　　b) $K_{1,8}$　　c) $K_{4,4}$　　d) C_7　　e) W_7　　f) Q_4

在练习 13～17 里，确定图是否为偶图。



18. 对哪些 n 值来说下列图是偶图？

 a) K_n b) C_n c) W_n d) Q_n

19. 下列图有多少个顶点和多少条边？

 a) K_n b) C_n c) W_n d) $K_{m,n}$ e) Q_n

20. 若图有度为 4，3，3，2，2 的顶点，则它有多少条边？

21. 是否存在带有下列度的有 5 个顶点的图？若有，则画出这样的图。

 a) 3，3，3，3，2 b) 1，2，3，4，5 c) 1，2，3，4，4

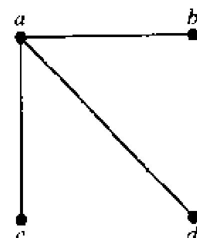
 d) 3，4，3，4，3 e) 0，1，2，2，3 f) 1，1，1，1，1

22. 至少带有 1 个顶点的 K_2 的子图有多少个？

23. 至少带有 1 个顶点的 K_3 的子图有多少个？

24. 至少带有 1 个顶点的 W_3 的子图有多少个？

25. 画出右面这个图的所有子图。



26. 设 G 是带有 v 个顶点和 e 条边的图。设 M 是 G 的顶点的最大度，m 是 G 的顶点的最小度。证明：

 a) $2e/v \geqslant m$ b) $2e/v \leqslant M$

若简单图每个顶点的度都相等，则这个图称为**正则的**。若正则图每个顶点的度都为 n，则这个图称为 n **正则的**。

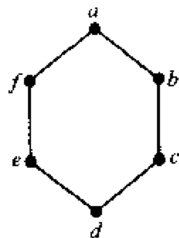
27. 对哪些 n 值来说下列图是正则图？

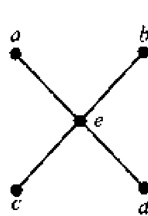
 a) K_n b) C_n c) W_n d) Q_n

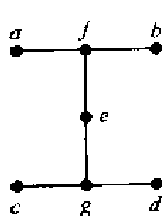
28. 对哪些 m 和 n 的值来说 $K_{m,n}$ 是正则图？

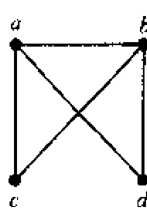
29. 度为 4 而且带有 10 条边的正则图有多少个顶点？

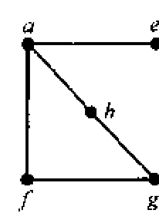
在练习 30～32 里，求给定的简单图对的并图。（假设带有相同端点的边是相同的。）

30.




31.




32.


33. 简单图 G 的补图$\overline{G}$与 G 有相同的顶点。两个顶点在$\overline{G}$里相邻，当且仅当它们在 G 里不相邻。求下列图。

 a) $\overline{K_n}$ b) $\overline{K_{m,n}}$ c) $\overline{C_n}$ d) $\overline{Q_n}$

34. 若 G 是有 15 条边的简单图而且$\overline{G}$有 13 条边，则 G 有多少个顶点？

35. 若简单图 G 有 v 个顶点和 e 条边，则$\overline{G}$有多少条边？

*36. 证明：若 G 是有 v 个顶点和 e 条边的偶简单图，则 $e \leqslant v^2/4$。

37. 证明：若 G 是有 n 个顶点的简单图，则 G 和 $\overline{G}$的并图是 K_n。

*38. 描述判定图是否为偶图的算法。

表示成 G^c 的有向图 $G=(V,E)$ 的逆图是图 (V,F)，其中 $(u,v)\in F$，当且仅当 $(v,u)\in E$。

39. 描述 7.1 节练习 7～9 里每个图的逆图。
40. 证明：每当 G 是有向图时，就有 $(G^c)^c=G$。
41. 证明：图 G 是它自身的逆图，当且仅当 G 所关联的关系（见 6.3 节）是对称的。
42. 把有向图的逆图的定义推广到有向多重图的逆图的概念。
43. 画出互联 9 个并行处理器的栅格网络。
44. 在互联 $n=m^2$ 个处理器的变种的栅格网络里，处理器 $P(i,j)$ 连接 4 个处理器 $P((i\pm1)\bmod m,j)$，$P(i,(j\pm1)\bmod m)$，使得连接沿栅格的边卷绕。画出 16 个处理器的这种变种的栅格网络。
45. 证明：在 $n=m^2$ 个处理器栅格网络里用 $O(\sqrt{n})=O(m)$ 个 hop 就能让每一对处理器互相通信。

7.3　图的表示和图的同构

7.3.1　引言

存在许多种有用的方式可以对图进行表示。在本章里将看到设法选择最方便的表示是有助于对图的处理的。在本节将要说明如何用多种不同的方式来表示图。

有时两个图在这样一种意义下恰好具有相同的形式，就是在两个图的顶点之间存在着一一对应，这个对应保持边的对应关系。在这种情形下就说两个图是*同构的*。确定两个图是否同构，这是本节里将要研究的重要图论问题。

7.3.2　图的表示

表示不带多重边的图的一种方式是列出这个图的所有边。另一种表示不带多重边的图的方式是用*相邻表*，它规定与图的每个顶点相邻的顶点。

例 1　用相邻表描述图 7-25 所给出的简单图。

解　表 7-2 列出与图的每个顶点相邻的顶点。■

例 2　通过列举在图的每个顶点上发出的边的终点，来表示图 7-26 所示的有向图。

解　表 7-3 表示图 7-26 所示的有向图。■

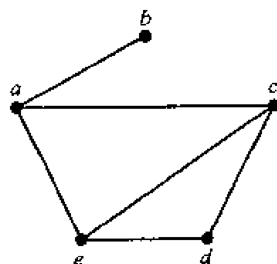


图 7-25　一个简单图

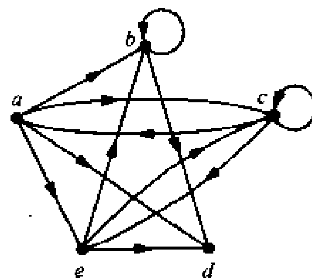


图 7-26　一个有向图

表 7-2 一个简单图的边表

顶　点	相邻顶点
a	b，c，e
b	a
c	a，d，e
d	c，e
e	a，c，d

表 7-3 一个有向图的边表

起　点	终　点
a	b，c，d，e
b	b，d
c	a，c，e
d	
e	b，c，d

7.3.3 相邻矩阵

若图里有许多边，则把图表示成边表或相邻表，就不便于执行图的算法。为了简化计算，可用矩阵表示图。在这里将给出两种类型的常用的表示图的矩阵。一种类型是基于顶点的相邻关系，另一种类型是基于顶点与边的关联关系。

假设$G=(V,E)$是简单图，其中$|V|=n$。假设把 G 的顶点任意地排列成 v_1，v_2，…，v_n。对这个顶点表来说，G 的相邻矩阵 $\mathbf{A}$（或 $\mathbf{A}_G$）是一个 $n\times n$ 的 0－1 矩阵，它满足这样的性质：当 v_i 和 v_j 相邻时第（i，j）项是 1，当 v_i 和 v_j 不相邻时第（i，j）项是 0。换句话说，若相邻矩阵是 $\mathbf{A}=[a_{ij}]$，则

$$a_{ij}=\begin{cases}1, & \text{若 } \{v_i, v_j\} \text{ 是 } G \text{ 的一条边} \\ 0, & \text{否则}\end{cases}$$

注意图的相邻矩阵依赖于所选择的顶点的顺序。因此带 n 个顶点的图有 $n!$ 个不同的相邻矩阵，因为 n 个顶点有 $n!$ 个不同的顺序。

简单图的相邻矩阵是对称的，即 $a_{ij}=a_{ji}$，这是因为当 v_i 和 v_j 相邻时，这两个项都是 1，否则都是 0。另外，因为简单图无环，所以每一项 a_{ii} 都是 0，$i=1$，2，3，…，n。

注意，当图里的边相对少时，相邻矩阵是稀疏矩阵，即只有很少的非 0 项的矩阵。可以用特殊的方法来表示和计算这样的矩阵。另外，当表示和处理这样的图时，用边表有时是更有效的。

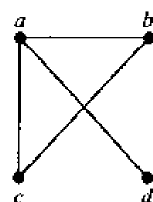


图 7-27 一个简单图

例 3 用相邻矩阵表示图 7-27 所示的图。

解 把顶点排列成 a，b，c，d。表示这个图的矩阵是

$$\begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}$$

■

例 4 画出带有相对于顶点顺序 a，b，c，d 的相邻矩阵的图。

$$\begin{bmatrix}0 & 1 & 1 & 0\\ 1 & 0 & 0 & 1\\ 1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0\end{bmatrix}$$

解 图 7-28 显示带有这个相邻矩阵的图。■

相邻矩阵也可用来表示带环和多重边的无向图。把顶点 a_i 上的环表示成相邻矩阵第 (i, i) 位置上的 1。当出现多重边时，相邻矩阵不再是 0－1 矩阵，这是因为相邻矩阵的第 (i, j) 项等于与 $\{a_i, a_j\}$ 关联的边数。包括多重图与伪图在内的所有无向图都具有对称的相邻矩阵。

例 5 用相邻矩阵表示图 7-29 所示的伪图。

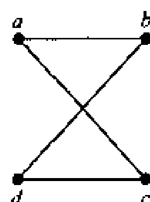


图 7-28 带有给定的相邻矩阵的图

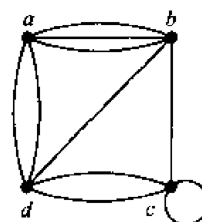


图 7-29 一个伪图

解 顶点顺序为 a，b，c，d 的相邻矩阵是

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$ ■

在第 6 章里用 0－1 矩阵表示有向图。若有向图 $G=(V,E)$ 从 v_i 到 v_j 有边，则它的矩阵在 (i,j) 位置上有 1，其中 v_1，v_2，…，v_n 是任意的顶点顺序。换句话说，若 $\mathbf{A}=[a_{ij}]$ 是相对于这个顶点表的相邻矩阵，则

$$a_{ij}=\begin{cases}1, & 若\ (v_i,\ v_j)\ 是\ G\ 的一条边 \\ 0, & 否则\end{cases}$$

有向图的相邻矩阵不必是对称的，因为当从 a_i 到 a_j 有边时，从 a_j 到 a_i 可以没有边。

相邻矩阵也可用来表示有向多重图。同样，当有连接两个顶点的同向多重边时，这样的矩阵不是 0－1 矩阵。在有向多重图的相邻矩阵里，a_{ij} 等于关联到 $(v_i,\ v_j)$ 的边数。

7.3.4 关联矩阵

表示图的另一种常用方式是用关联矩阵。设 $G=(V,E)$ 是无向图。设 v_1，v_2，…，v_n 是顶点而 e_1，e_2，…，e_m 是边。则相对于 V 和 E 的这个顺序的关联矩阵是 $n\times m$ 矩阵 $\mathbf{M}=[m_{ij}]$，其中

$$m_{ij}=\begin{cases}1, & 当边\ e_j\ 关联\ v_i\ 时 \\ 0, & 否则\end{cases}$$

例 6 用关联矩阵表示图 7-30 所示的图。

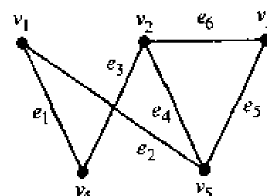


图 7-30 无向图

解 关联矩阵是

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

■

关联矩阵也可用来表示多重边和环。在关联矩阵里用相等项的列来表示多重边，因为这些边关联同样一对顶点。用恰有一项等于1的列来表示环，它对应于环所关联的顶点。

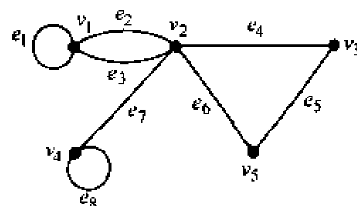


图 7-31 伪图

例 7 用关联矩阵表示图 7-31 所示的伪图。

解 这个图的关联矩阵是

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

■

7.3.5 图的同构

经常需要知道是否有可能以同样的方式来画出两个图。例如，在化学里用图为化合物建模。不同的化合物可能分子式相同但结构不同。把这样的化合物表示成不能以同样方式来画的图。用表示过去已知化合物的图来判定想象中的新化合物是否已经研究过了。

对具有同样结构的图来说，存在着一些有用的术语。

定义 1 设 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$ 是简单图，若存在一对一的和映上的从 V_1 到 V_2 的函数 f，且 f 具有这样的性质，对 V_1 里所有的 a 和 b 来说，a 和 b 在 G_1 里相邻，当且仅当 $f(a)$ 和 $f(b)$ 在 G_2 里相邻，就说 G_1 与 G_2 是同构的。这样的函数 f 称为同构㊀。

换句话说，当两个简单图同构时，两个图的顶点之间具有保持相邻关系的一一对应。

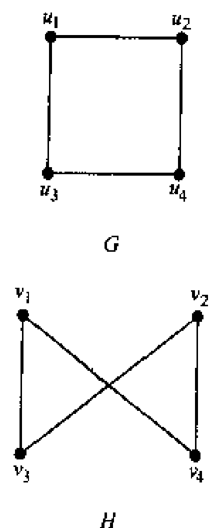


图 7-32 图 G 和 H

例 8 证明图 7-32 所示图 $G=(V,E)$ 和 $H=(W,F)$ 同构。

㊀ 同构（isomorphism）这个词来自两个希腊语字根：表示“相等”的 isos 和表示“形式”的 morphe。

解 函数 f 满足 $f(u_1)=v_1$，$f(u_2)=v_4$，$f(u_3)=v_3$，$f(u_4)=v_2$，它是 V 和 W 之间的一一对应。为了看出这个对应保持相邻关系，注意 G 里相邻的顶点是 u_1 和 u_2，u_1 和 u_3，u_2 和 u_4，以及 u_3 和 u_4，由 $f(u_1)=v_1$ 和 $f(u_2)=v_4$，$f(u_1)=v_1$ 和 $f(u_3)=v_3$，$f(u_2)=v_4$ 和 $f(u_4)=v_2$，以及 $f(u_3)=v_3$ 和 $f(u_4)=v_2$ 所组成的每一对顶点都是在 H 里相邻的。■

判定两个简单图是否同构常常是一件困难的事情。在两个带有 n 个顶点的简单图的顶点集之间有 $n!$ 种可能的一一对应。若 n 太大，则通过检验每一种对应来看它是否保持相邻关系和不相邻关系，这样做是不可行的。

不过，常常可以通过说明两个简单图不具有同构的图所必须具有的性质来说明它们不同构。把这样的性质称为对简单图的同构来说的**不变量**。例如，同构的简单图必然具有相同的顶点数，因为在这些图的顶点集之间有着一一对应。而且，同构的简单图必然有相同的边数，因为在顶点之间的一一对应建立了边之间的一一对应。另外，同构的简单图必然有相同的顶点度。即 G 里的 d 度顶点 v 必然对应 H 里的 d 度顶点 $f(v)$，这是因为在 G 里顶点 w 与 v 相邻，当且仅当在 H 里 $f(v)$ 与 $f(w)$ 相邻。

例 9 说明图 7-33 所示的图不同构。

解 G 和 H 都具有 5 个顶点和 6 条边。不过，G 有 1 度顶点 e 而 H 没有 1 度顶点。所以 G 与 H 是不同构的。■

顶点数、边数以及顶点度都是在同构下的不变量。若两个简单图的这些量有任何不同，则这两个图就不是同构的。不过，当这些量都相同时，也不一定意味着两个图是同构的。目前还没有已知的用来判定简单图是否同构的不变量集。

例 10 判定图 7-34 所示的图是否同构。

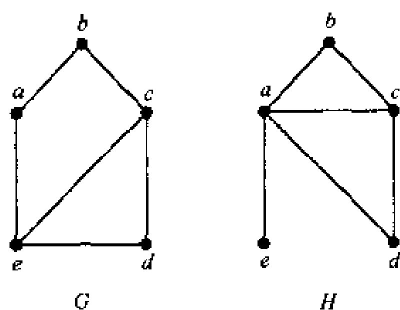


图 7-33 图 G 和 H

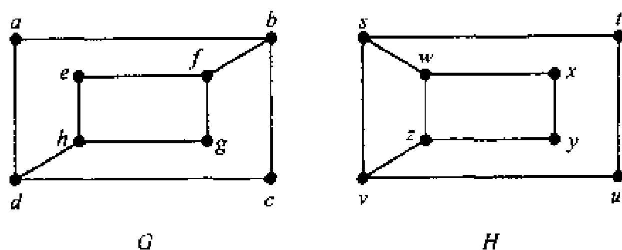


图 7-34 图 G 和 H

解 图 G 和 H 都具有 8 个顶点和 10 条边。它们都具有 4 个 2 度顶点和 4 个 3 度顶点。因为这些不变量都相同，所以可能想到它们是同构的。

然而 G 和 H 不是同构的。为了看明白这一点，注意到因为在 G 里 $\deg(a)=2$，所以 a 必然对应于 H 里的 t，u，x 或 y，这是因为这些顶点是 H 里的 2 度顶点。不过，H 里的这 4 个顶点中每一个都与 H 里另一个 2 度顶点相邻，但是在 G 里 a 却不是这样的。

看出 G 与 H 不同构的另一种方式是，注意到若这两个图同构，则由 3 度顶点和连接它们的边所组成的子图同构（读者应当验证它）。然而图 7-35 所示的这些子图却不是同构的。■

为了说明从图 G 的顶点集到图 H 的顶点集的函数 f 是一个同构，需要说明 f 保持边的关系。一种有助于这样做的方式是利用相邻矩阵。具体地说，为了说明 f 是一个同构，就说明 G 的相邻矩阵与 H 的相邻矩阵相同，其中 G 的相邻矩阵的行和列的标记都是 G 里的顶点，H 的相邻矩阵的行和列的标记都是 G 里的对应顶点在 f 下的像。在下一个例子里解释如何这样做。

例 11　判定图 7-36 所示的图 G 和 H 是否同构。

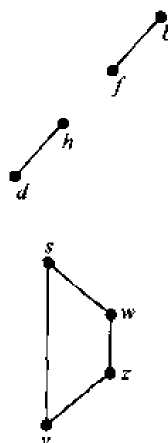


图 7-35　由 3 度顶点和连接它们的边所组成的 G 和 H 的子图

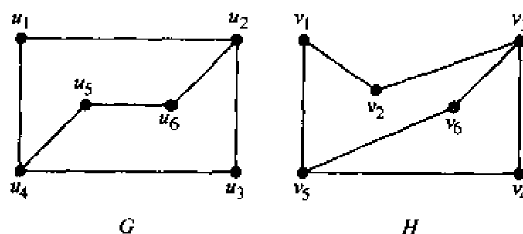


图 7-36　图 G 和 H

解　G 和 H 都具有 6 个顶点和 7 条边，都具有 4 个 2 度顶点和 2 个 3 度顶点。还容易看出由 2 度顶点和连接它们的边所组成的 G 和 H 的子图是同构的（读者应当验证它）。因为 G 和 H 对这些不变量来说是相同的，这就有理由试着找出一个同构 f。

现在定义函数 f，然后判定它是否同构。因为 $\deg(u_1)=2$ 而且 u_1 不与任何其他 2 度顶点相邻，所以 u_1 的像必然是 v_4 或 v_6，它们是 H 里仅有的不与 2 度顶点相邻的顶点。随意地令 $f(u_1)=v_6$。［如果发现这个选择得不出同构，就接着试验 $f(u_1)=v_4$。］因为 u_2 与 u_1 相邻，所以 u_2 可能的像是 v_3 和 v_5。随意地令 $f(u_2)=v_3$。照这样继续下去，用顶点的相邻关系和度作为指引，令 $f(u_3)=v_4$，$f(u_4)=v_5$，$f(u_5)=v_1$，以及 $f(u_6)=v_2$。现在已经有了在 G 的顶点集与 H 的顶点集之间的一一对应，即 $f(u_1)=v_6$，$f(u_2)=v_3$，$f(u_3)=v_4$，$f(u_4)=v_5$，$f(u_5)=v_1$，以及 $f(u_6)=v_2$。为了查看 f 是否保持边，就检查 G 的相邻矩阵

$$
\mathbf{A}_G = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array}
\begin{array}{c} \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array} \\
\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}
$$

以及用 G 里的对应顶点的像来标记行和列的 H 的相邻矩阵

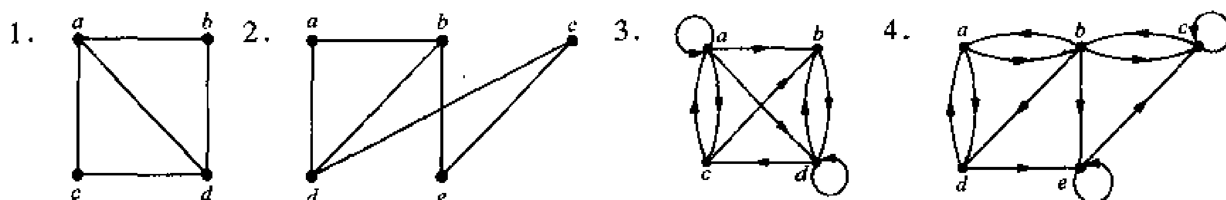
$$
\mathbf{A}_H = \begin{array}{c} \\ v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{array}
\begin{array}{c} \begin{array}{cccccc} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{array} \\
\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}
$$

因为 $\mathbf{A}_G = \mathbf{A}_H$，所以 f 是保持边的。由此得出 f 是一个同构，所以 G 与 H 是同构的。注意，若事实证明 f 不是一个同构，则并没有证明 G 与 H 不是同构的，因为 G 和 H 里的顶点的另一个对应仍然可以是同构。■

判定两个图是否同构，已知的最好算法具有指数的最坏情形时间复杂性（对图的顶点数来说）。不过，已经知道了解决这个问题的线性平均情形时间复杂性的算法，而且有希望找到判定两个图是否同构的多项式最坏情形时间复杂性的算法。一种名为 NAUTY 的最好的实用算法，可用来在现代个人电脑上在 1 秒钟之内判定带有 100 个顶点的两个图是否同构。可以在因特网上下载 NAUTY 软件并用它做实验。

练习

在练习 1~4 中，用相邻表表示给定的图。



5. 用相邻矩阵表示练习 1 中的图。
6. 用相邻矩阵表示练习 2 中的图。
7. 用相邻矩阵表示练习 3 中的图。
8. 用相邻矩阵表示练习 4 中的图。
9. 用相邻矩阵表示下列每一个图。

 a) K_4　b) $K_{1,4}$　c) $K_{2,3}$　d) C_4　e) W_4　f) Q_3

在练习 10~12 中，画出带有给定的相邻矩阵的图。

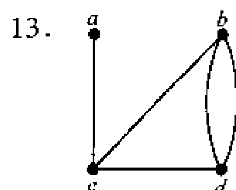
10. $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

11. $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

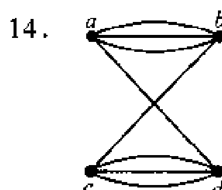
12. $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

在练习 13~15 中，用相邻矩阵表示所给定的图。

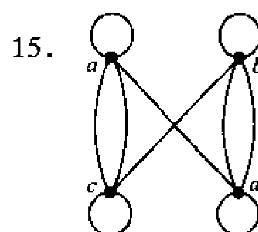
13.



14.



15.



在练习 16~18 中，画出用给定的相邻矩阵所表示的无向图。

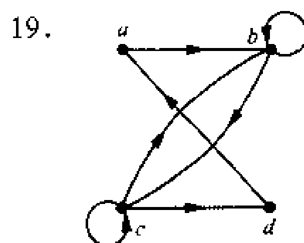
16. $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$

17. $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

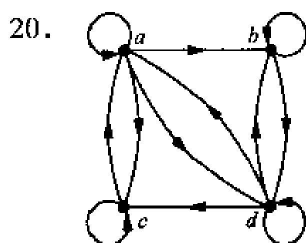
18. $\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$

在练习 19~21 中，求给定的有向图的相邻矩阵。

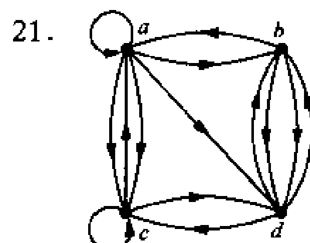
19.



20.



21.



在练习 22~24 中，画出用给定的相邻矩阵所表示的图。

22. $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

23. $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$

24. $\begin{bmatrix} 0 & 2 & 3 & 0 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$

25. 每一个对称的和对角线全为 0 的 0－1 方阵是否都是简单图的相邻矩阵？

26. 用关联矩阵表示练习 1 和 2 中的图。

27. 用关联矩阵表示练习 13～15 中的图。

*28. 什么是无向图的相邻矩阵的一行中的各项之和？对有向图来说呢？

*29. 什么是无向图的相邻矩阵的一列中的各项之和？对有向图来说呢？

30. 什么是无向图的关联矩阵的一行中的各项之和？

31. 什么是无向图的关联矩阵的一列中的各项之和？

*32. 求下列每个图的相邻矩阵。

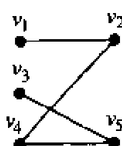
a) K_n　　b) C_n　　c) W_n　　d) $K_{m,n}$　　e) Q_n

*33. 求练习32 a)－d)中的图的关联矩阵。

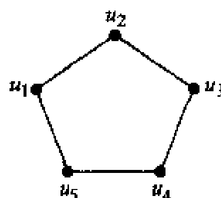
在练习 34～44 里，判定所给定的一对图是否同构。

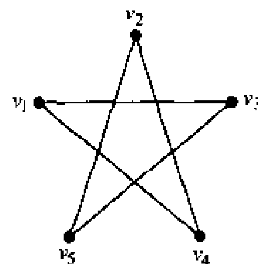
34.



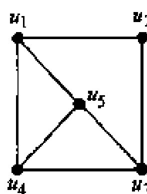


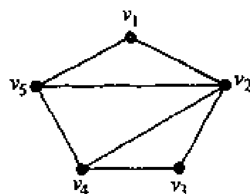
35.



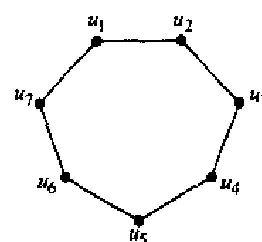


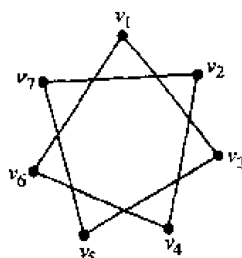
36.



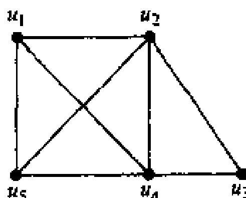


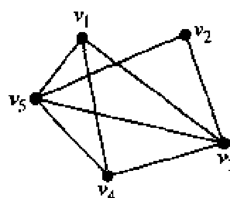
37.



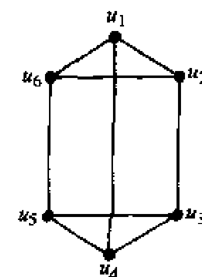


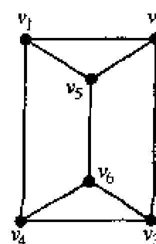
38.



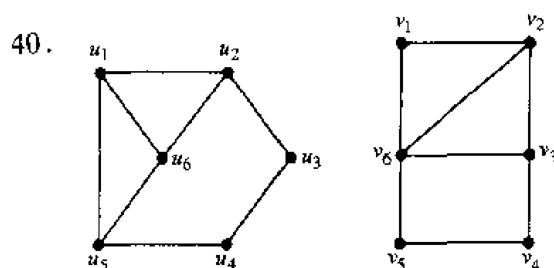


39.

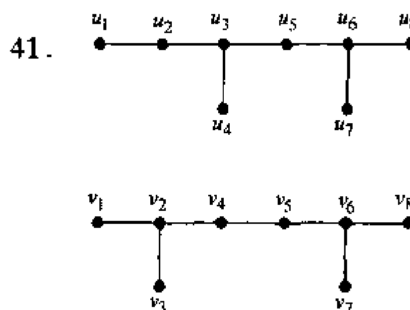




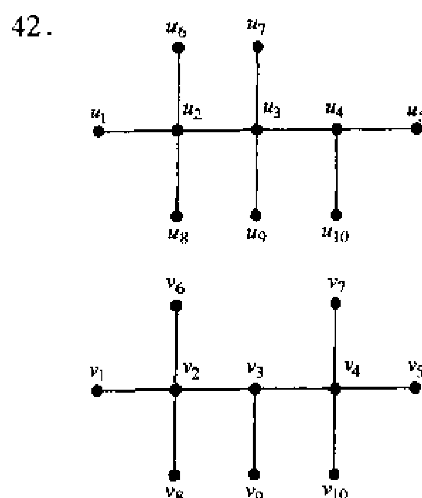
40.



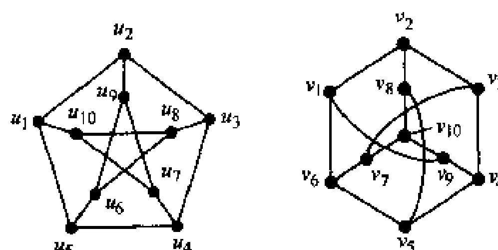
41.



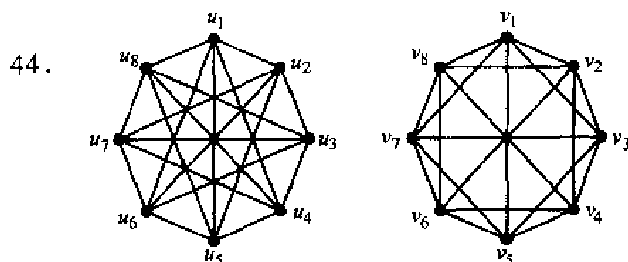
42.



43.



44.



45. 证明：简单图的同构关系是等价关系。

46. 设 G 和 H 是同构的简单图。证明：它们的补图 $\overline{G}$ 和 $\overline{H}$ 也是同构的。

47. 描述一下图的相邻矩阵里对应于孤立点的行和列。

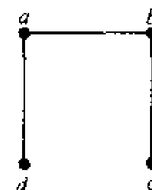
48. 描述一下图的关联矩阵里对应于孤立点的行。

49. 证明：带有 2 个以上顶点的偶图的顶点可以排序，使得相邻矩阵形如

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{B} & \mathbf{0} \end{bmatrix}$$

其中所示的四项都是矩形块。

若简单图 G 和 $\overline{G}$ 是同构的，则 G 称为自补图。



50. 证明：右图是自补图。

51. 求带有 5 个顶点的自补简单图。

*52. 证明：若 G 是带有 v 个顶点的自补简单图，则 $v \equiv 0$ 或 $1 \pmod 4$。

53. 对哪些整数 n 来说 C_n 是自补图？

54. 带有 n 个顶点的非同构的简单图有多少个？其中 n 是

a) 2　　b) 3　　c) 4

55. 带有 5 个顶点和 3 条边的非同构的简单图有多少个？

56. 带有 6 个顶点和 4 条边的非同构的简单图有多少个？

57. 带有下列相邻矩阵的简单图是否同构？

a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

58. 判定带有下列关联矩阵的无环图是否同构的。

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

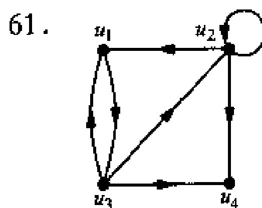
b) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

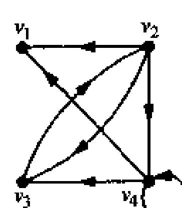
59. 把简单图的同构定义推广到包含环和多重边的无向图。

60. 定义有向图的同构。

在练习 61～64 中，判定所给定的一对图是否同构。

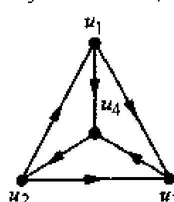
61.

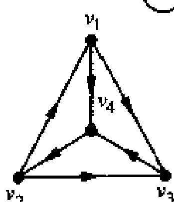




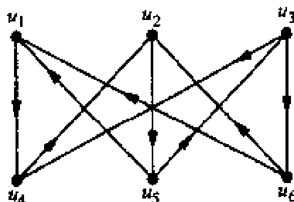
62.

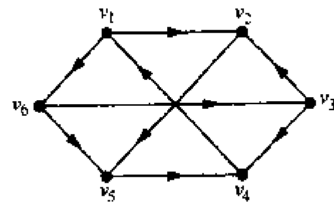
63.





64.





65. 证明：若 G 和 H 是同构的有向图，则 G 和 H 的逆（在 7.2 节练习 39 之前定义）也是同构的。

*66. 带有 n 个顶点的非同构有向图有多少个？其中 n 是

a) 2 b) 3 c) 4

*67. 无向图的关联矩阵与它的转置之积是什么？

*68. 表示带有 v 个顶点和 e 条边的简单图需要多少存储空间？其中分别利用

a) 相邻表 b) 相邻矩阵 c) 关联矩阵

对所谓的同构检验来说，魔鬼对是该检验不能证明为不同构的一对不同构的图。

69. 求一个用于检验的魔鬼对，检查两个图里的顶点度的序列确定它们是相同的。

7.4 连通性

7.4.1 引言

许多问题可以用沿图的边旅行所形成的通路来建模。例如，判定在两个计算机之间用中间连接能否传递消息的问题，就可以用图模型来研究。利用由图里的通路所组成的模型，可以解决投递邮件、收集垃圾以及计算机网络中的诊断等的路线有效计划问题。

7.4.2 通路

首先定义描述通路的基本的图论术语。

定义 1 在无向图里从 u 到 v 的长度为 n 的通路是由图的边所组成的序列 $e_1, \ldots, e_n$，使得 $f(e_1)=\{x_0,x_1\}$，$f(e_2)=\{x_1,x_2\}$，…，$f(e_n)=\{x_{n-1},x_n\}$，其中 n 是正整数，$x_0=u$ 而 $x_n=v$。当这个图是简单图时，就用顶点序列 $x_0, x_1, \ldots, x_n$ 表示这条通路(这是因为列出这些顶点就唯一地确定了通路)。若一条通路在相同的顶点上开始和结束，即 $u=v$，则它是一条回路。把这条通路或回路说成是经过或行遍顶点 $x_1, x_2, \ldots, x_{n-1}$。若通路或回路不重复地包含相同的边，则它是简单的。

当没有必要区分多重边时，就用顶点序列 $x_0, x_1, \ldots, x_n$ 表示通路 $e_1, e_2, \ldots e_n$，其中对 $i=1, 2, \ldots, n$ 来说 $f(e_i)=\{x_{i-1},x_i\}$。这种记法仅仅指出通路所经过的顶点。经过这个顶点序列可以有多条通路。

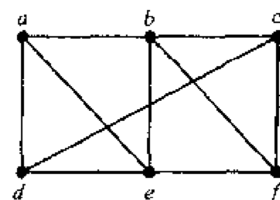


图 7-37 简单图

例 1 在图 7-37 所示的图里，a, d, c, f, e 是长度为 4 的简单通路，因为 $\{a, d\}$，$\{d, c\}$，$\{c, f\}$ 和 $\{f, e\}$ 都是边。但是 d, e, c, a 不是通路，因为 $\{e, c\}$ 不是边。注意 b, c, f, e, b 是长度为 4 的回路，这是因为 $\{b, c\}$，$\{c, f\}$，$\{f, e\}$ 和 $\{e, b\}$ 都是边，这条通路在 b 上开始和结束。长度为 5 的通路 a, b, e, d, a, b 不是简单的，因为它两次包含边 $\{a, b\}$。 ■

有向图里的通路和回路在第 6 章里介绍过。现在定义对有向多重图来说的这样的通路。

定义 2 在有向图里从 u 到 v 的长度为 n 的通路，是由图的边所组成的序列 $e_1, \ldots, e_n$，使得 $f(e_1)=(x_0,x_1)$，$f(e_2)=(x_1,x_2)$，…，$f(e_n)=(x_{n-1},x_n)$，其中 n 是正整数，$x_0=u$ 而 $x_n=v$。当图里没有多重边时，就用顶点序列 $x_0, x_1, \ldots, x_n$ 表示这条通路。把在相同顶点上开始和结束的通路称为回路或圈。若一条通路或回路不重复地包含相同的

边，则把它称为简单的。

当没有必要区分多重边时，就用顶点序列 x_0，x_1，…，x_n 表示通路 e_1，e_2，…，e_n，其中对 $i=1$，2，…，n 来说 $f(e_i)=(x_{i-1},x_i)$。这种记法仅仅指出通路所经过的顶点。经过这个顶点序列可以有多条通路。

7.4.3 无向图连通性

若消息通过一个或多个中间计算机来传递，则何时计算机网络具有每对计算机都可共享信息的性质？当利用图来表示这个计算机网络，其中用顶点表示计算机而用边表示通信连接时，这个问题就变成：何时在图里任何两个顶点之间都存在通路？

定义 3　若无向图每一对不同的顶点之间都有通路，则该图称为*连通的*。

因此在网络里，任何两个计算机都可以通信，当且仅当这个网络的图是连通的。

例 2　图 7-38 里的图 G 是连通的，这是因为在每一对不同的顶点之间都有通路（读者应当验证它）。但是图 7-38 里的图 H 不是连通的。例如，在顶点 a 和 d 之间没有通路。■

在第 8 章里将需要下述的定理。

定理 1　在连通无向图的每一对不同顶点之间都存在简单通路。

证　设 u 和 v 是连通无向图 $G=(V,E)$ 的两个不同的顶点。因为 G 是连通的，所以 u 和 v 之间至少有 1 条通路。设 x_0，x_1，…，x_n 是长度最短的通路的顶点序列，其中 $x_0=u$ 而 $x_n=v$。这条长度最短的通路是简单的。为了看明白这一点，假设它不是简单的。则对满足 $0\leqslant i<j$ 的某个 i 和 j 来说有 $x_i=x_j$。这意味着通过删除顶点序列 x_i，…，x_{j-1} 所对应的边，就获得了带有顶点序列 x_0，x_1，…，x_{i-1}，x_j，…，x_n 的从 u 到 v 的更短通路。□

不连通的图是 2 个或 2 个以上连通子图之并，每一对子图都没有公共的顶点。这些不相交的连通子图称为图的*连通分支*。

例 3　图 7-39 所示的图 G 的连通分支是什么？

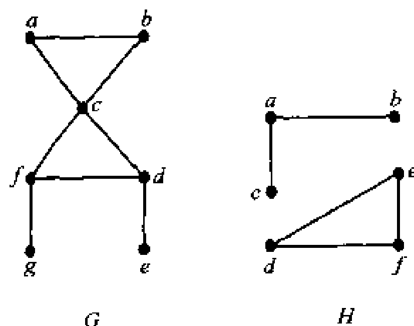


图 7-38　图 G 和 H

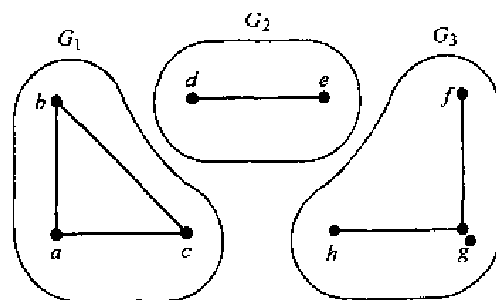


图 7-39　图 G 和它的连通分支 G_1，G_2 和 G_3

解　图 G 是图 7-39 所示的 3 个不相交连通子图 G_1，G_2 和 G_3 之并。这 3 个子图是 G 的连通分支。■

有时删除一个顶点和它所关联的边，就产生带有比原图更多的连通分支的子图。把这样的顶点称为割点（或节点）。从连通图里删除割点，就产生不连通的子图。同理，把一旦删除就产生带有比原图更多的连通分支的子图的边称为割边或桥。

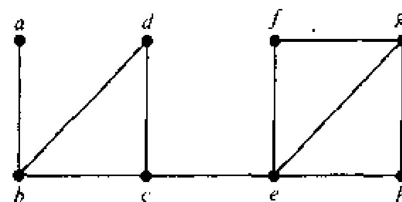


图 7-40　图 G

例 4　求出图 7-40 所示的图 G 的割点和割边。

解　图 G 的割点是 b，c 和 e。删除这些顶点中的一个（和它的邻边），就使得这个图不再是连通的。割边是 $\{a, b\}$ 和 $\{c, e\}$。删除这些边中的一条，就使得 G 不再是连通的。■

7.4.4　有向图中的连通性

根据是否考虑边的方向，在有向图里有两种连通性概念。

定义 4　若每当 a 和 b 都是一个有向图里的顶点时，就有从 a 到 b 和从 b 到 a 的通路，则该图是**强连通的**。

为了让一个有向图是强连通的，从这个图里任何一个顶点到任何另外一个顶点必需存在有向边的序列。有向图可以不是强连通的但还是"一整块"。为了准确地说明这一点，给出下述的定义。

定义 5　若在有向图的底图里，任何两个顶点之间都有通路，则该有向图是**弱连通的**。

即有向图是弱连通的，当且仅当在忽略边的方向时，任何两个顶点之间总是存在通路。显然，任何强连通有向图也是弱连通的。

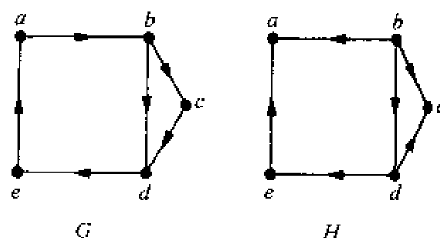


图 7-41　有向图 G 和 H

例 5　图 7-41 所示的有向图 G 和 H 是否强连通的？是否弱连通的？

解　G 是强连通的，因为在这个有向图里，任何两个顶点之间都存在通路（读者应当验证它）。因此 G 也是弱连通的。图 H 不是强连通的。在这个图里，从 a 到 b 没有有向通路。但是 H 是弱连通的，因为在 H 的底图里，任何两个顶点之间都有通路（读者应当验证它）。■

7.4.5　通路与同构

有多种方式可以利用通路和回路来帮助判定两个图是否同构的。例如，具有特定长度的简单回路的存在性，就是一种可以用来证明两个图是不同构的有用的性质。另外，利用通路来构造可能是同构的映射。

曾经指出过，简单图的一个有用的同构不变量是长度为 k 的简单回路的存在性，其中 k 是大于 2 的正整数。（这个不变量的证明在本章末尾留作练习 36。）例 6 说明如何用这个不变量来证明两个图不同构。

例 6　判定图 7-42 所示的图 G 和 H 是否同构。

解　G 和 H 都具有 6 个顶点和 8 条边。各自具有 4 个 3 度顶点和 2 个 2 度顶点。所以对两个图来说有 3 个不变量（顶点数、边数以及顶点度）都是相同的。但是 H 有长度为 3 的简单回路。即 v_1，v_2，v_6，v_1，而 G 没有长度为 3 的简单回路（G 里的所有简单回路的长度至少为 4）。因为长度为 3 的简单回路的存在性是一个同构不变量，所以 G 和 H 不是同构的。■

已经说明了某种类型的通路，即具有特定长度的简单回路，是如何用来证明两个图是不同构的。还可以用通路来求出潜在的同构映射。

例 7　判定图 7-43 所示的图 G 和 H 是否同构。

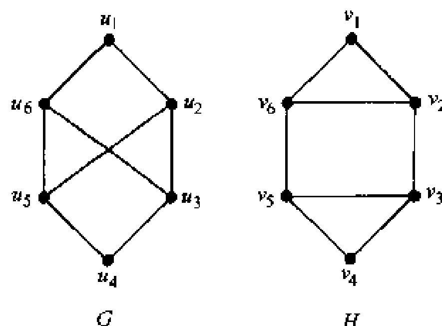


图 7-42　图 G 和 H

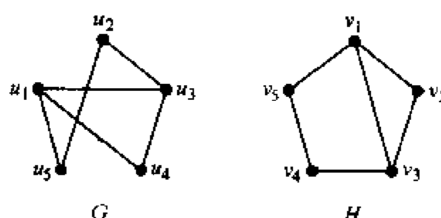


图 7-43　图 G 和 H

解　G 和 H 都具有 5 个顶点和 6 条边，都具有 2 个 3 度顶点和 3 个 2 度顶点，都具有 1 个长度为 3 的简单回路，1 个长度为 4 的简单回路，以及 1 个长度为 5 的简单回路。因为所有这些同构不变量都是相同的，所以 G 和 H 可能是同构的。为了求出可能的同构，跟随经过所有顶点并且使得两图里的对应顶点的度都相同的通路。例如，G 里的通路，u_1，u_4，u_3，u_2，u_5 和 H 里的通路 v_3，v_2，v_1，v_5，v_4 都经过图里的每一个顶点，都在 3 度顶点上开始，都分别经过 2、3 和 2 度顶点并且在 2 度顶点上结束。通过在图里跟随这些通路，定义映射 f 满足 $f(u_1)=v_3$，$f(u_4)=v_2$，$f(u_3)=v_1$，$f(u_2)=v_5$ 和 $f(u_5)=v_4$。通过说明 f 保持边，或者通过说明在顶点的适当顺序下 G 和 H 的相邻矩阵是相同的，读者就可以说明 f 是一个同构，所以 G 与 H 是同构的。■

7.4.6　统计顶点之间的通路

在一个图里两个顶点之间的通路的数目，可以用这个图的相邻矩阵来确定。

定理 2　设 G 是带有相对于顶点顺序 v_1，v_2，…，v_n 的相邻矩阵 **A** 的图（允许带有向边或无向边、带多重边和环）。从 v_i 到 v_j 的长度为 r 的不同通路的数目等于 $\mathbf{A}^r$ 的第 (i, j) 项，其中 r 是正整数。

证　用数学归纳法证明。设 G 是带有相邻矩阵 **A** 的图（假设 G 的顶点具有顺序 v_1，v_2，…，v_n）。从 v_i 到 v_j 长度为 1 的通路数是 **A** 的第 (i, j) 项，这是因为该项是从 v_i 到 v_j 的边数。

假设 $\mathbf{A}^r$ 的第 (i, j) 项是从 v_i 到 v 的长度为 r 的不同通路的个数。这是归纳假设。因

为 $\mathbf{A}^{r+1}=\mathbf{A}^r\mathbf{A}$，所以 $\mathbf{A}^{r+1}$的第（i，j）项等于

$$b_{i1}a_{1j}+b_{i2}a_{2j}+\cdots+b_{in}a_{nj}$$

其中 b_{ik}是 $\mathbf{A}^r$ 的第（i，k）项。根据归纳假设，b_{ik}是从 v_i 到 v_k 的长度为 r 的通路数。

从 v_i 到 v_j 的长度为 $r+1$ 的通路，包括从 v_i 到某个中间顶点 v_k 的长度为 r 的通路以及从 v_k 到 v_j 的边。根据计数的乘法规则，这样的通路的个数是从 v_i 到 v_k 的长度为 r 的通路数（即 b_{ik}）与从 v_k 到 v_j 的边数（即 a_{kj}）之积。当对所有可能的中间顶点 v_k 求这些乘积之和时，根据计数的加法规则，就得出所需要的结果。■

例 8 在图 7-44 所示的简单图 G 里，从 a 到 d 的长度为 4 的通路有多少条？

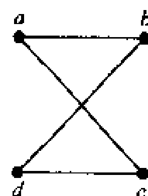


图 7-44 图 G

解 G 的相邻矩阵（顶点顺序为 a，b，c，d）是

$$\mathbf{A}=\begin{bmatrix}0&1&1&0\\1&0&0&1\\1&0&0&1\\0&1&1&0\end{bmatrix}$$

因此从 a 到 d 的长度为 4 的通路数是 $\mathbf{A}^4$ 的第（1，4）项。因为

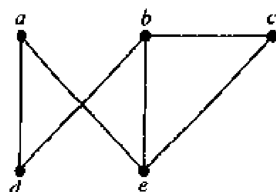
$$\mathbf{A}^4=\begin{bmatrix}8&0&0&8\\0&8&8&0\\0&8&8&0\\8&0&0&8\end{bmatrix}$$

所以恰好有 8 条从 a 到 d 的长度为 4 的通路。通过检查这个图，我们看出 a，b，a，b，d；a，b，a，c，d；a，b，d，b，d；a，b，d，c，d；a，c，a，b，d；a，c，a，c，d；a，c，d，b，d 和 a，c，d，c，d 是 8 条从 a 到 d 的通路。■

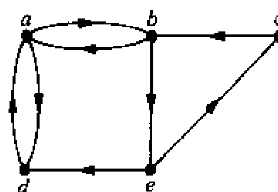
定理 2 可以用来求出在图的两个顶点之间的最短通路的长度（见练习 32），还可以用来判定图是否连通（见练习 37 和 38）。

练习

1. 下述的每个顶点表是否可以构成下图里的通路？哪些通路是简单的？哪些是回路？那些通路的长度是多少？
   a) a,e,b,c,b b) a,e,a,d,b,c,a
   c) e,b,a,d,b,e d) c,b,d,a,e,c

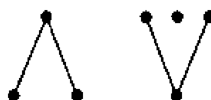


2. 下述的每个顶点列表是否可以构成下图里的通路？哪些通路是简单的？哪些是回路？那些通路的长度是多少？
   a) a,b,e,c,b b) a,d,a,d,a
   c) a,d,b,e,a d) a,b,e,c,b,d,a

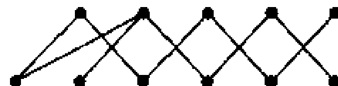


在练习 3～5 中，判定所给的两个图是否连通。

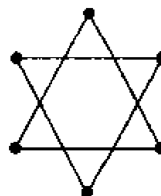
3.



4.



5.



6. 在练习 3～5 中，每个图各自有多少个连通分支？对每个图来说，求出它的每个连通分支。

*7. 求出 K_4 中两个不同顶点之间长度为 n 的通路的数目，若 n 是

a) 2　b) 3　c) 4　d) 5

*8. 对练习 7 中的 n 值来说，求出 $K_{3,3}$里任何两个相邻顶点之间长度为 n 的通路的数目。

*9. 对练习 7 中的 n 值来说，求出 $K_{3,3}$里任何两个不相邻顶点之间长度为 n 的通路的数目。

10. 求出在图 7-37 的图中 c 和 d 之间具有如下长度的通路的数目。

a) 2　b) 3　c) 4　d) 5　e) 6　f) 7

11. 求出在练习 2 的有向图中从 a 到 e 具有如下长度的通路的数目。

a) 2　b) 3　c) 4　d) 5　e) 6　f) 7

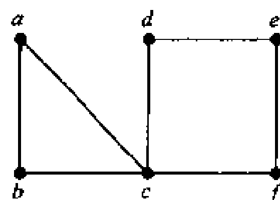
*12. 证明：带有 n 个顶点的连通图至少具有 $n-1$ 条边。

13. 设 $G=(V,E)$是简单图。设 R 是 V 上的关系，它是由这样的顶点对(u,v)所组成的，使得存在从 u 到 v 的通路、或使得 $u=v$。证明：R 是一个等价关系。

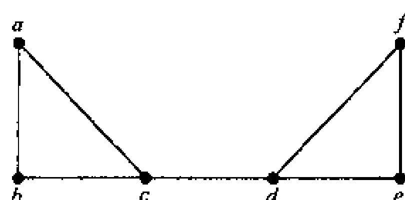
*14. 证明：在任何简单图中，从任何奇数度顶点都有通路到某个其他的奇数度顶点。

在练习 15～17 中，求所给图的所有割点。

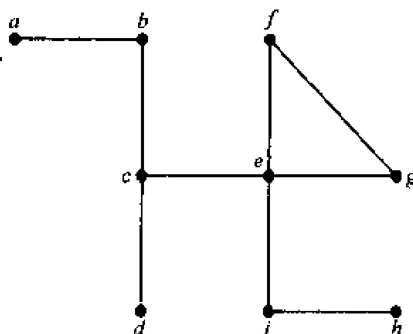
15.



16.



17.



18. 求练习 15～17 中的图的所有割边。

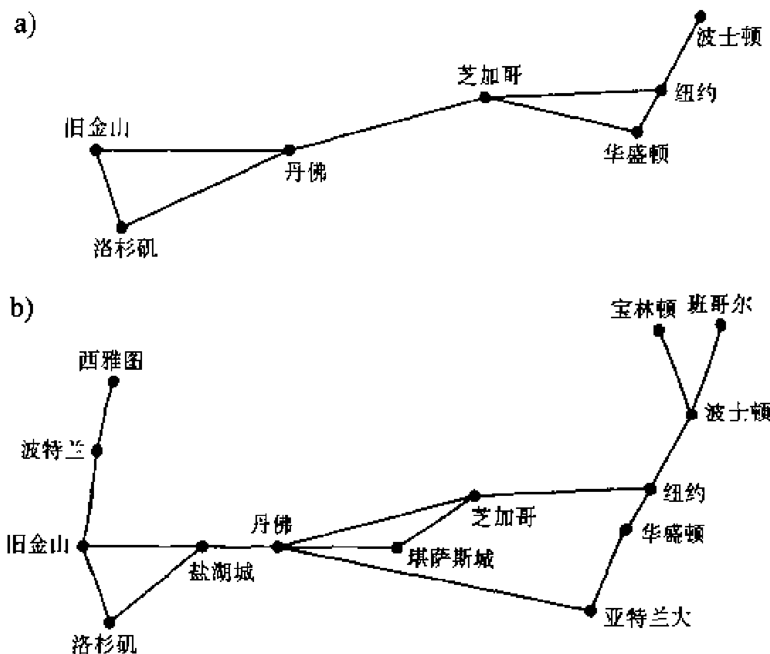
*19. 假设 v 是一条割边的端点。证明：v 是割点当且仅当它不是悬挂点。

*20. 证明：在连通简单图 G 中，顶点 c 是割点当且仅当存在着与 c 不同的顶点 u 和 v，使得在 u 和 v 之间的每一条通路都经过 c。

*21. 证明：具有至少 2 个顶点的简单图至少有 2 个顶点不是割点。

*22. 证明：简单图中的一条边是割边当且仅当它不属于该图里任何一条简单回路。

23. 若网络里的通信连接故障会导致不能传送某些消息，则应当提供备份连接。对下面 a)和 b)里所示的通信网络来说，确定出那些应当有备份的连接。



一个有向图中的顶点基是顶点的一个集合，从该集合中的某个顶点到有向图中不在集合中的任何顶点都有一条通路，并且从集合中任何顶点到该集合中任何另一个顶点都没有通路。

24. 对 7.2 节练习 7～9 中的每个有向图求顶点基。

25. 在影响图（在 7.1 节例 2 里描述）中顶点基的重要性是什么？找出该例里的影响图的顶点基。

☞ 26. 证明：若连通有向图 G 是图 G_1 和 G_2 的并图，则 G_1 和 G_2 至少具有 1 个公共的顶点。

*27. 证明：若简单图 G 有 k 个连通分支，而且这些分支分别具有 n_1，n_2，…，n_k 个顶点，则 G 的边数不超过

$$\sum_{i=1}^{k} C(n_i,2)$$

*28. 用练习 27 证明：带有 n 个顶点和 k 个连通分支的简单图最多有 $(n-k)(n-k+1)/2$ 条边。[提示：先证明

$$\sum_{i=1}^{k} n_i^2 \leqslant n^2-(k-1)(2n-k)$$

其中 n_i 是第 i 个连通分支的顶点数。]

*29. 证明：若带有 n 个顶点的简单图 G 具有超过 $(n-1)(n-2)/2$ 条边，则它是连通的。

30. 当把每个连通分支中的顶点都连续地列出时，描述一下带有 n 个连通分支的图的相邻矩阵。

31. 存在多少个不同构的带有 n 个顶点的连通简单图？其中 n 是

　a) 2　　b) 3　　c) 4　　d) 5

32. 解释一下如何用定理 2 求出图里从顶点 v 到顶点 w 的最短通路的长度。

33. 用定理 2 求出图 7-37 中的多重图从 a 到 f 的最短通路的长度。

34. 用定理 2 求出练习 2 中的有向图从 a 到 c 的最短通路的长度。

☞35. 设 P_1 和 P_2 是简单图 G 中顶点 u 和 v 之间的没有相同边的两条简单通路。证明：在 G 中存在简单回路。

36. 证明：长度为 k 的简单回路的存在性是一个同构不变量，其中 k 是大于 2 的正整数。

37. 解释一下如何用定理 2 判定一个图是否连通。

38. 用练习 37 证明：图 7-38 中的图 G 是连通的而图 H 不是连通的。

7.5 欧拉通路与哈密顿通路

7.5.1 引言

普鲁士的哥尼斯堡镇（现名加里宁格勒，属于俄罗斯共和国）被普雷格尔河支流分成 4 部分。这 4 部分包括河岸两部分，河中心岛以及两条支流之间的部分。在 18 世纪有 7 座桥连接这 4 部分。图 7-45 画出了这些部分和桥。

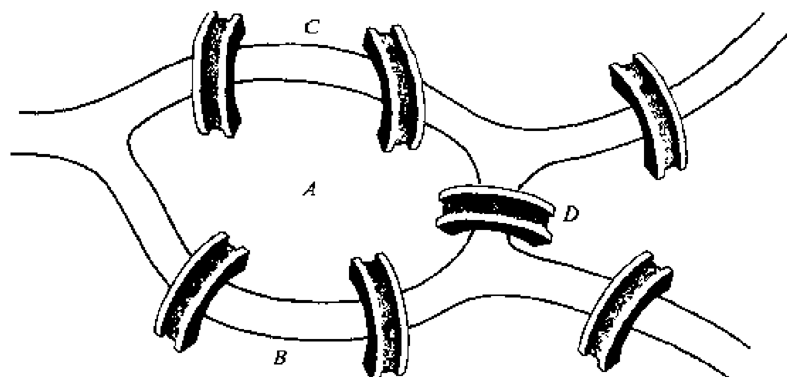


图 7-45 哥尼斯堡七桥

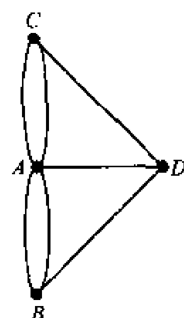


图 7-46 哥尼斯堡镇的多重图模型

镇上的人们在周日里穿过镇子进行长距离的散步。他们想弄明白是否可能从镇里某个位置出发不重复地经过所有桥并且返回出发点。

瑞士数学家列昂哈德·欧拉[⊖]解决了这个问题。他的解答在 1736 年发表，这也许是第一次使用图论。欧拉利用多重图来研究这个问题，其中用顶点表示 4 个部分，用边表示桥。图 7-46 显示这个多重图。

⊖ 列昂哈德·欧拉（Leonhard Euler，1707—1783） 欧拉是瑞士巴塞尔附近一位加尔文教派牧师之子。他 13 岁进入巴塞尔大学，遵照父亲的愿望开始神学生涯。在大学里，欧拉受到著名的伯努利数学家族的约翰·伯努利的指导。他的兴趣和技巧使他放弃神学研究而转向数学。欧拉 16 岁取得哲学硕士学位。1727 年他得到大帝邀请他加入圣彼得堡的科学院。1741 年他来到柏林科学院，在这里一直呆到 1766 年。然后他回到圣彼得堡，并在那里度过余生。

欧拉令人难以置信地多产，对数学的许多领域作出了贡献，包括数论、组合数学、分析以及在诸如音乐和造船学这样的领域的应用等。他写了数量在 1100 以上的书籍和文章，而且留下了如此多的未发表的著作，以致于在他去世之后，用了 47 年才发表完他的所有著作。在他活着的时候，他的文章积累得如此快，使得他总有一大摞文章等待发表。柏林科学院先发表了这一摞顶上的文章，所以后来的结果常常先于它们所依赖或取代的结果而发表。欧拉有 13 个孩子，当有一两个孩子在他膝上玩耍时，他能够照样工作。在他生命的最后 17 年里，他完全失明了，但是由于他奇妙的记忆力，这并没有影响他的数学产出。他的全集的出版工作由瑞士自然科学协会负责，目前还在进行之中，预期将超过 75 卷。

不重复地经过每一座桥旅行的问题可以利用这个模型来重新叙述。问题变成：在这个多重图里是否存在着包含每一条边的简单回路？

定义 1　图 G 里的欧拉回路是包含着 G 的每一条边的简单回路。图 G 里的欧拉通路是包含着 G 的每一条边的简单通路。

下述的例子解释了欧拉回路和欧拉通路的概念。

例 1　在图 7-47 里，哪些无向图具有欧拉回路？在没有欧拉回路的那些图里，哪些具有欧拉通路？

解　图 G_1 具有欧拉回路，例如 a, e, c, d, e, b, a。G_2 和 G_3 都没有欧拉回路（读者应当验证它）。但是 G_3 具有欧拉通路，即 a, c, d, e, b, d, a, b。G_2 没有欧拉通路（读者应当验证它）。■

例 2　在图 7-48 里，哪些有向图具有欧拉回路？在没有欧拉回路的那些图里，哪些具有欧拉通路？

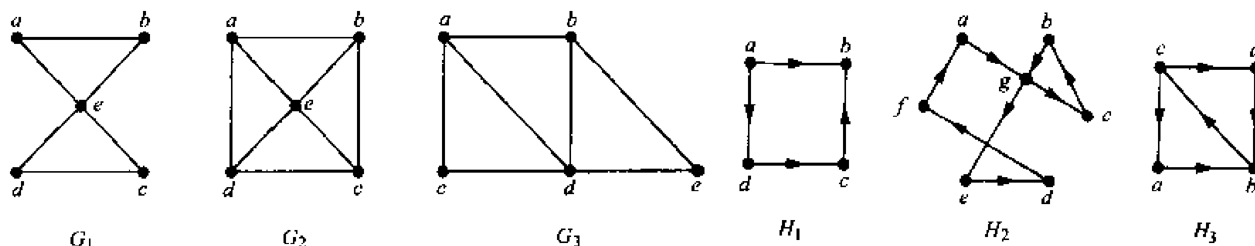


图 7-47　无向图 G_1，G_2 和 G_3

图 7-48　有向图 H_1，H_2 和 H_3

解　图 H_2 具有欧拉回路，例如 $a, g, c, b, g, e, d, f, a$。$H_1$ 和 H_3 都没有欧拉回路（读者应当验证它）。H_3 具有欧拉通路，即 c, a, b, c, d, b，但是 H_1 没有欧拉通路（读者应当验证它）。■

7.5.2　欧拉回路和欧拉通路的充要条件

对判断多重图是否具有欧拉回路和欧拉通路来说，存在着简单的标准。欧拉在解决著名的哥尼斯堡七桥问题时发现了它们。假设在本节里讨论的所有图都具有有穷多个顶点和边。

若一个连通多重图具有欧拉回路，则能说出它的什么性质来呢？可以说明的是：每一个顶点都必有偶数条边。为此，首先注意一条欧拉回路从顶点 a 开始，接着是 a 关联的一条边，比方说 $\{a, b\}$。边 $\{a, b\}$ 为 deg (a) 贡献 1 度。这条回路每次经过一个顶点就为该顶点贡献 2 度，这是因为这条回路经过关联该顶点的边进入，又经过另一条这样的边离开。最后，这条回路在它开始的地方结束，为 deg (a) 贡献 1 度。因此 deg (a) 必为偶数，这是因为当回路开始时它贡献 1 度，当回路结束时它贡献 1 度，每次经过 a 都贡献 2 度（如果它经过了 a）。除 a 外的其余顶点都有偶数度，这是因为每次回路经过一个顶点就为该顶点贡献 2 度。由此得出结论，若连通图有欧拉回路，则每一个顶点必有偶数度。

欧拉回路存在的这个必要条件是否也是充分的？即若在连通多重图里所有顶点都有偶数度，则是否必有欧拉回路？这个问题可以通过构造来肯定地解决。

假设 G 是连通多重图而且 G 的每一个顶点都有偶数度。构造从 G 的任意顶点 a 开始的简单回路。设 $x_0=a$。首先任意地选择一条关联 a 的边 $\{x_0, x_1\}$。通过建立尽量长的简单通路 $\{x_0, x_1\}$，$\{x_1, x_2\}$，…，$\{x_{n-1}, x_n\}$ 来继续构造。例如，在图 7-49 的图 G 中，从 a 开始而且连续地选择边 $\{a, f\}$，$\{f, c\}$，$\{c, b\}$ 和 $\{b, a\}$。

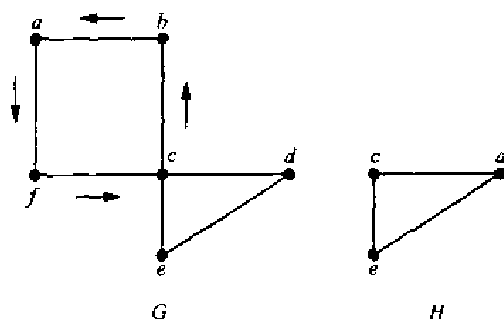


图 7-49　构造 G 中的欧拉回路

这样的通路必然结束，这是因为图的边数是有穷的。它在 a 上以形如 $\{a, x\}$ 的边开始，而且在 a 上以形如 $\{y, a\}$ 的边结束。这是因为通路每次经过一个偶数度顶点时，它只用 1 条边进入这个顶点，所以至少还剩下 1 条边让通路离开这个顶点。这条通路可能用完所有的边，也可能没有用完。

若所有的边都已经用完，则欧拉回路已经构造好了。否则，考虑通过从 G 中删除已经用过的边和不关联任何剩余的边的顶点，这样得到子图 H。当从图 7-49 中删除回路 a，f，c，b，a 时，就得到标记为 H 的子图。

因为 G 是连通的，所以 H 与已经删除的回路至少具有 1 个公共顶点。设 w 是这样的顶点。(在这个例子里，c 是这个顶点。)

H 中的每一个顶点都有偶数度（因为 G 中的所有顶点都有偶数度，对每个顶点来说，把与这个顶点关联的边成对地删除，以便形成 H)。注意 H 可能是不连通的。像在 G 中做过的那样，在 w 上开始，通过尽可能地选择边来构造 H 中的简单回路。这条回路必然在 w 上结束。例如，在图 7-49 中，c，d，e，c 是 H 中的回路。下一步通过把 H 中的回路与 G 中原来的回路拼接起来形成 G 中的回路（这是可行的，因为 w 是这个回路的顶点之一)。当在图 7-49 中这样做时，就得到回路 a，f，c，d，e，c，b，a。

继续进行这个过程，直到已经用完了所有的边为止。(这个过程必然结束，这是因为图中只有有穷的边数。)这样就产生出欧拉回路。这样的构造说明，若连通多重图的顶点都有偶数度，则该图具有欧拉回路。

把这些结果总结成定理 1。

定理 1　连通多重图具有欧拉回路当且仅当它的每个顶点都有偶数度。

现在可以解决哥尼斯堡七桥问题了。因为图 7-46 所示的表示这些桥的多重图具有 4 个奇数度顶点，所以它没有欧拉回路。无法在给定点开始，恰好经过每座桥一次，并返回开始点。

算法 1 给出了在定理 1 前面的讨论中所给的求欧拉回路的构造过程。(因为这个过程中的回路是任意地选择的，所以存在一些不确定性。不打算通过更精确地说明过程的步骤来消除这些不确定性。)

下一个例子说明如何利用欧拉通路和欧拉回路来解决一种智力题。

例 3　有许多智力题要求用铅笔连续移动，不离开纸面并且不重复地画出图形。利用欧拉回路和欧拉通路来解决这样的智力题。例如，能否用这样的方法画出图 7-50 所示的穆罕默德短弯刀？其中图形是在同一个顶点上开始和结束。

算法 1 构造欧拉回路

```
procedure Euler(G：所有顶点有偶数度的连通多重图)
circuit :=在 G 中任选的顶点开始连续地加入边所形成的回到该顶点的回路
H :=删除这条回路的边之后的 G
while H还有边
begin
    subcircuit :=在既是 H 中的顶点也是circuit 一条边的端点开始的 H 中的一条回路
    H :=删除 subcircuit 的边和所有孤立点之后的 H
    circuit :=在适当顶点上插入 subcircuit 之后的 circuit
end {circuit 是欧拉回路}
```

解 可以解决这个问题，因为图 7-50 所示的图 G 具有欧拉回路。它具有这样的回路是因为它的所有顶点偶有偶数度。用算法 1 来构造欧拉回路。首先形成回路 a，b，d，c，b，e，i，f，e，a。通过删除这条回路的边并且删除因此产生的孤立点，就得到子图 H。然后形成 H 中的回路 d，g，h，j，i，h，k，g，f，d。形成这条回路之后就用完了 G 中的所有边。在适当的地方拼接这条回路和第一条回路，就产生出欧拉回路 a，b，d，g，h，j，i，h，k，g，f，d，c，b，e，i，f，e，a。这条回路给出了铅笔不离开纸面并且不重复地画出弯刀的方法。■

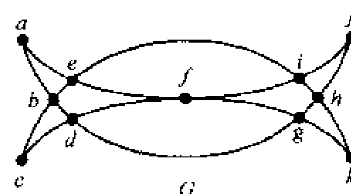


图 7-50 穆罕默德短弯刀

构造欧拉回路的另一个算法称为弗勒里算法，在本节末尾的习题里描述它。

现在说明，连通多重图具有欧拉通路（不是欧拉回路）当且仅当它恰有两个奇数度顶点。首先假设连通多重图有从 a 到 b 的欧拉通路，但不是欧拉回路。通路的第一条边为 a 贡献 1 度。通路每次经过 a 就为 a 贡献 2 度。通路的最后一条边为 b 贡献 1 度。通路每次经过 b 就为 b 贡献 2 度。所以 a 和 b 的度都是奇数。其他每一个顶点都具有偶数度，这是因为每当通路经过一个顶点时，就为这个顶点贡献 2 度。

现在考虑相反的情况。假设这个图恰有两个奇数度顶点，比方说 a 和 b。考虑由原来的图和边 $\{a, b\}$ 所组成的更大的图。这个更大的图的每一个顶点都有偶数度，所以具有欧拉回路。删除新边就产生原图的欧拉通路。下述的定理总结了这些结果。

定理 2 连通多重图具有欧拉通路而无欧拉回路，当且仅当它恰有两个奇数度顶点。

例 4 图 7-51 所示的哪些图具有欧拉通路？

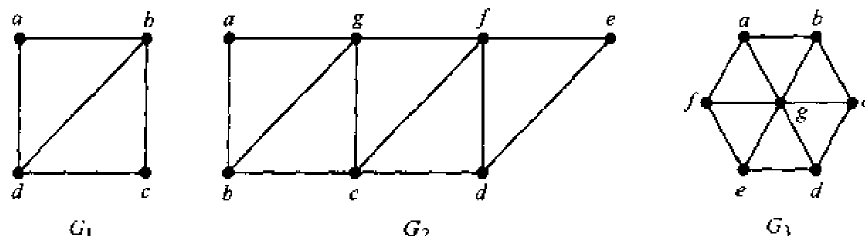


图 7-51 三个无向图

解 G_1 恰有两个奇数度顶点，即 b 和 d。因此它具有必须用 b 和 d 作为端点的欧拉通

路。一条这样的欧拉通路是 d，a，b，c，d，b。同理，G_2 恰有两个奇数度顶点，即 b 和 d。因此它具有必须用 b 和 d 作为端点的欧拉通路。一条这样的欧拉通路是 b，a，g，f，e，d，c，g，b，c，f，d。G_3 没有欧拉通路，因为它具有 6 个奇数度顶点。■

回到 18 世纪的哥尼斯堡，是否有可能在镇里某点开始，经过所有的桥，在镇里其他某点结束？通过判定表示哥尼斯堡七桥的多重图是否具有欧拉通路，就可以回答这个问题。因为这个多重图有四个奇数度顶点，所以没有欧拉通路，这样的旅行是不可能的。

有向图中欧拉通路和欧拉回路的充要条件，在本节末尾的练习里讨论。

7.5.3 哈密顿通路和回路

已经讨论了包含多重图每一条边恰好一次的通路和回路存在的充要条件。能否对包含图的每一个顶点恰好一次的简单通路和回路做同样的事情？

定义 2 在图 $G=(V,E)$ 中，若 $V=\{x_0,x_1,\cdots,x_{n-1},x_n\}$ 并且对 $0\leqslant i<j\leqslant n$ 来说有 $x_i\neq x_j$，则通路 x_0，x_1，…，x_{n-1}，x_n 称为哈密顿通路。在图 $G=(V,E)$ 中，若 x_0，x_1，…，x_{n-1}，x_n 是哈密顿通路，则 x_0，x_1，…，x_{n-1}，x_n，x_0（其中 $n>1$）称为哈密顿回路。

这个术语来自爱尔兰数学家威廉·罗万·哈密顿爵士[⊖]在 1857 年发明的智力题。哈密顿的智力题包含木质十二面体［如图 7-52 a 所示，十二面体有十二个正五边形表面］、十二面体每个顶点上的钉子以及细线。十二面体的 20 个顶点用世界上不同城市作标记。智力题的目标是在一个城市开始，沿十二面体的边旅行，访问其他 19 个城市每个恰好一次，回到第一个城市结束。旅行经过的回路用钉子和细线来标记。

因为作者不可能向每位读者提供带钉子和细线的木质十二面体，所以考虑一个等价的问题：图 7-52 b 所示图中是否具有恰好经过每个顶点一次的回路？这就解决了这个智力题，因为该图与包含十二面体顶点和边的图同构。图 7-53 显示了哈密顿智力题的解。



⊖ 威廉·罗万·哈密顿（William Rowan Hamilton，1805—1865） 哈密顿这位以往最有名的爱尔兰科学家 1805 年出生在都柏林。他的父亲是成功的律师，母亲来自以智力超群而闻名的家族。他本人是个神童。到 3 岁时他就是一名出色的读者并掌握了高等算术。因为他的聪明，他被送到身为著名语言学家的叔叔詹姆士的身边。到 8 岁时哈密顿学会了拉丁语、希腊语和希伯来语；到 10 岁时他又学会了意大利语和法语，并且开始学习东方语言，包括阿拉伯语、梵语和波斯语。在此期间他以懂得当时的所有语言而自豪。17 岁时他不再学习新的语言，但是已经掌握了微积分和许多数学天文学，他开始了在光学上的开创性工作，还发现了拉普拉斯的天体力学著作中的重大错误。哈密顿在 18 岁进入都柏林三一学院之前没有进过学校；他接受私人教育。在三一学院里他在科学和古典文学上都是表现超群的学生。在获得学位之前，他就因为才华出众被任命为爱尔兰皇家天文学家，战胜了争取这个职位的多位著名天文学家。他终身担任这个职位，在都柏林郊外的邓辛天文台生活和工作。哈密顿为光学、抽象代数和动力学作出了重要贡献。哈密顿发明了称为四元数的代数对象来作为非交换系统的例子。当他沿都柏林的运河散步时，发现了四元数相乘的适当方式。狂喜之下，他把公式刻在了跨越运河的石桥上，今日该地立匾为记。随后哈密顿一直沉迷在四元数里，努力把它们应用到数学的其他领域，而不再转向新的研究领域。

1857 年，哈密顿根据自己非交换代数的工作，发明了“艾口西安游戏”。他把这个想法以 25 镑的价格出售给游戏和智力题的经销商。（因为游戏的销路一直不好，事实证明这是经销商一次失败的投资。）本节所描述的智力题“旅行者十二面体”，又称“周游世界”，就是该游戏的变种。

哈密顿在 1833 年第三次结婚，但是他的婚姻结果很糟，因为他的妻子是半残疾人，无法处理他的家务。他在生命的最后 20 年里，过着酗酒和隐居的生活。他于 1865 年死于痛风，留下大批包含未发表过研究结果的文章。在这些文章里，混杂着大量晚餐碟子，许多碟子里还有已脱水的吃剩的排骨。

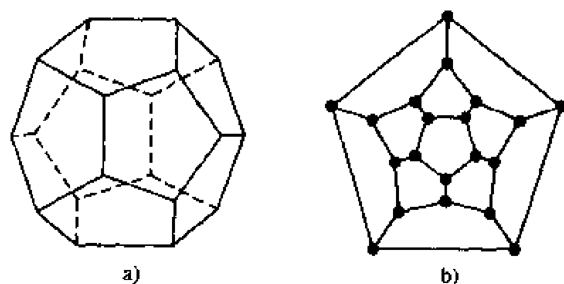


图 7-52　哈密顿"周游世界"的智力题

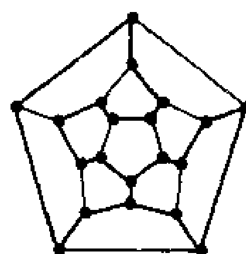


图 7-53　"周游世界"智力题的解

例 5　在图 7-54 中，哪些简单图具有哈密顿回路？或者没有哈密顿回路但是有哈密顿通路？

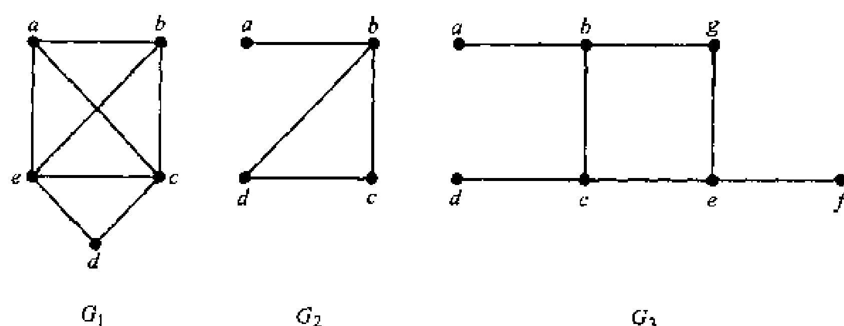


图 7-54　三个简单图

解　G_1 有哈密顿回路：a，b，c，d，e，a。G_2 没有哈密顿回路（注意，从包含每一个顶点的任何回路必然两次包含边 $\{a, b\}$，就可以看出这一点），但是 G_2 确实有哈密顿通路，即 a，b，c，d。G_3 既无哈密顿回路也无哈密顿通路，这是因为包含所有顶点的任何通路都必然多次包含边 $\{a, b\}$、$\{e, f\}$ 和 $\{e, d\}$ 其中之一。■

是否存在简单方式来判定一个图有无哈密顿回路或哈密顿通路？初看起来，似乎应当有判定这一点的简单方式，因为存在简单方式来回答一个图有无欧拉回路这样的相似问题。令人吃惊的是，没有已知的简单的充要条件来判定哈密顿回路的存在性。不过，已经知道有许多定理对哈密顿回路的存在性给出了充分条件。另外，某些性质可以用来证明一个图没有哈密顿回路。例如，带有 1 度顶点的图没有哈密顿回路，因为在哈密顿回路里每个顶点都关联着回路里的两条边。再有，若图中有 2 度顶点，则关联这个顶点的两条边属于任意一条哈密顿回路。还要注意，当构造哈密顿回路而且该回路经过某一个顶点时，除了回路所用到的两条边以外，不用再考虑这个顶点所关联的其他所有边。此外，哈密顿回路里不能包含更小的回路。

例 6　证明图 7-55 中的图都没有哈密顿回路。

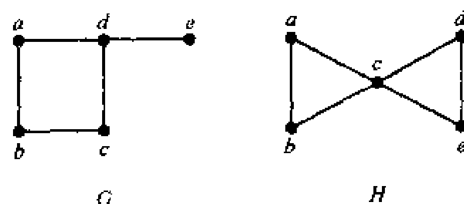


图 7-55　两个没有哈密顿回路的图

解　G 中没有哈密顿回路，因为 G 有 1 度顶点，即 e。现在考虑 H。因为顶点 a，b，d 和 e 的度都为 2，所以这些顶点关联的每一条边都必然属于任意一条哈密顿回路。现在容易看出 H 中不存在哈密顿回路，因为任何这样的哈密顿回路都不得不包含 4 条

关联 c 的边，这是不可能的。■

例 7　证明：每当 $n \geq 3$ 时 K_n 就有哈密顿回路。

解　从 K_n 中的任意一个顶点开始来形成哈密顿回路。以所选择的任意顺序来访问顶点，只要求通路在同一个顶点开始和结束，而且恰好访问其他每个顶点一次。这样做是可能的，因为在 K_n 中任意两个顶点之间都有边。■

现在陈述给出哈密顿回路存在的充分条件的定理。这只是已知的许多这样的定理之一。

定理 3　若 G 是带 n 个顶点的连通简单图，其中 $n \geq 3$，则 G 有哈密顿回路的充分条件是每个顶点的度都至少为 $n/2$。

已知的求一个图中的哈密顿回路或判定这样的回路不存在的最好算法具有指数级的最坏时间复杂性（相对于图的顶点数来说）。找到解决这个问题的具有多项式最坏时间复杂性的算法必定是一个重大成就，因为这样一个算法的存在将蕴含着许多其他似乎是难解的问题都可以用具有多项式的最坏时间复杂性的算法来解决。

现在给出哈密顿回路对编码的应用。

例 8　**格雷码**　旋转的指针的位置可以表示成数字的形式。一种方式是把圆周等分成 2^n 段弧并且用长度为 n 的位串给每段弧赋值。图 7-56 显示了用长度为 3 的位串来这样做的两种方式。

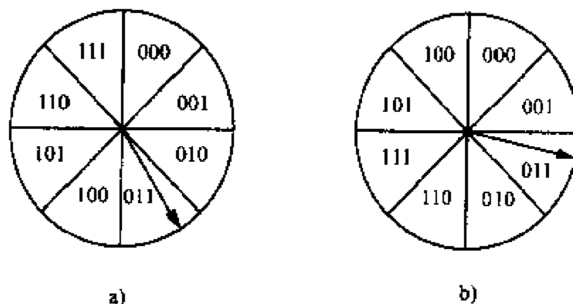


图 7-56　把指针位置转换成数字形式

用 n 个接触点来确定指针位置的数字表示。每个接触点用来读出位置的数字表示中的一位。图 7-57 针对图 7-56 中的两种赋值对此做了解释。

当指针靠近两段弧的边界时，在读出的指针位置中可能发生错误。这可能引起在读出的位串中的一个大的错误。例如，在图 7-56 a 的编码方案里，若在确定指针位置的过程中发生了一个小的错误，则读出的位串是 100 而不是 011。所有三位都是错的！为了把在确定指针位置的过程中的错误的影响降到最低，用位串对 2^n 段弧的赋值应当使相邻的弧所表示的位串只相差一位。在图 7-56 b 的编码方案中情况恰好就是这样。在确定指针位置的过程里的一个错误给出位串 010 而不是 011。只有一位是错的。

格雷码是圆周的弧的一种标记，使得相邻的弧具有恰好相差一位的位串标记。在图 7-56 b)里的赋值是一个格雷码。可以这样找出格雷码：以下述的方式列出所有长度为 n 的位串，使得每一个位串与前一个位串恰好相差一位，而且最后一个位串与第一个位串恰好相差一位。可以用 n 立方体 Q_n 来为这个问题建模。解决这个问题所需要的是 Q_n 中的一条哈密顿回路。这样的哈密顿回路容易求出。例如，Q_3 的一条哈密顿回路显示在图 7-58 中。这条哈密顿回路所产生的前后恰好相差一位的位串序列是 000，001，011，010，110，111，101 和 100。

格雷码是以弗兰克·格雷的名字命名的，在 20 世纪 40 年代，他在 AT&T 贝尔实验室工作，为了把传送数字信号过程中的错误的影响降到最低而发明了它们。■

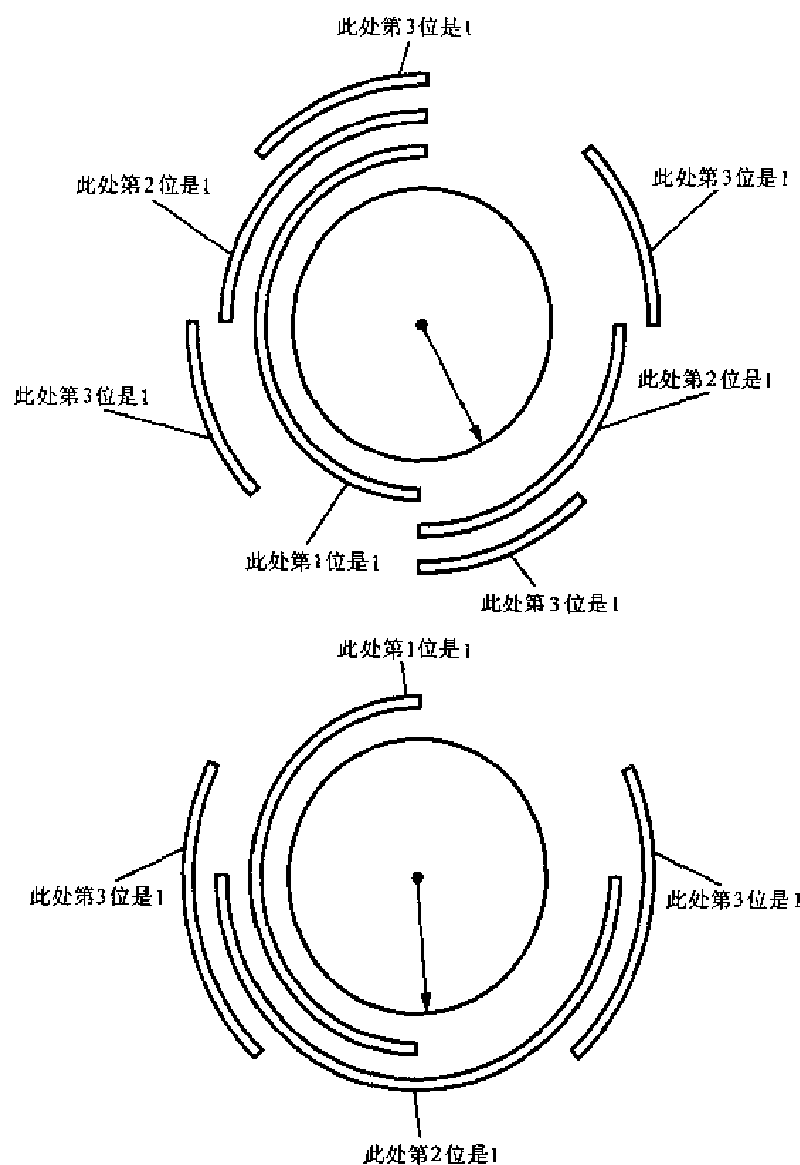


图 7-57　指针位置的数字表示

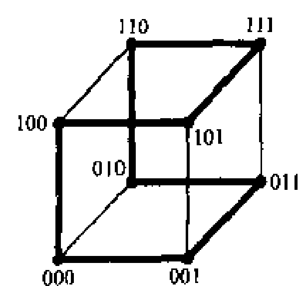
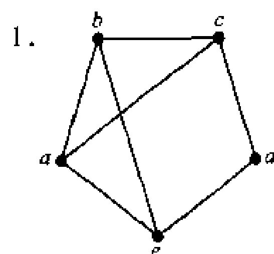
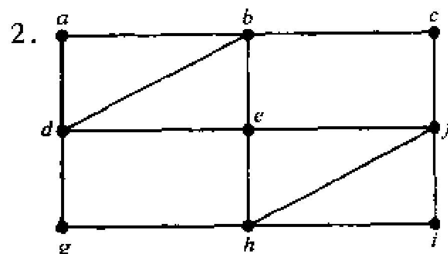


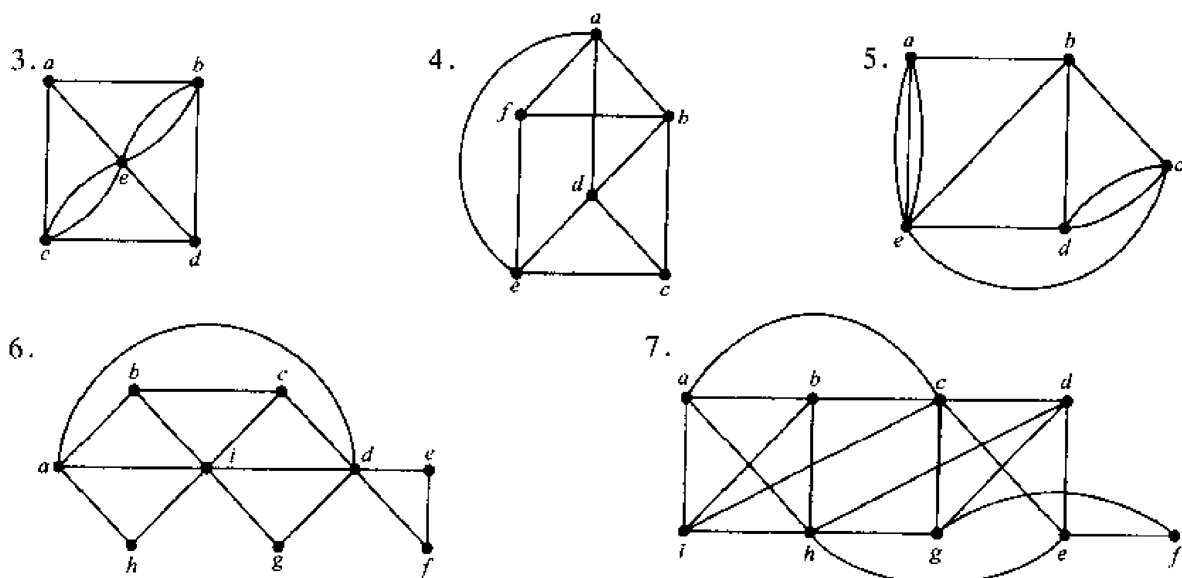
图 7-58　Q_3 的哈密顿回路

练习

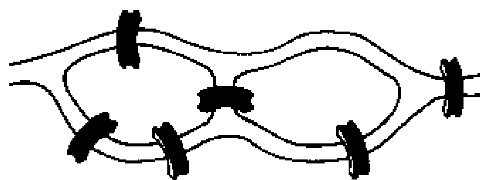
在练习 1～7 中，判定每个图是否具有欧拉回路。当存在欧拉回路时，构造出一条这样的回路。

1. 

2. 



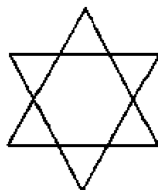
8. 判定练习 1 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
9. 判定练习 2 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
10. 判定练习 3 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
11. 判定练习 4 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
12. 判定练习 5 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
13. 判定练习 6 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
14. 判定练习 7 中的图是否具有欧拉通路。若存在欧拉通路，则构造出这样的一条通路。
15. 在加里宁格勒（哥尼斯堡的俄罗斯名称），除了 18 世纪的七座桥之外，还有另外两座桥。这些新桥分别连接区域 B 和 C 以及区域 B 和 D。是否有人能够经过加里宁格勒的九座桥恰好一次并且回到出发点？
16. 是否有人能够经过右图所示的所有桥恰好一次并且回到出发点？



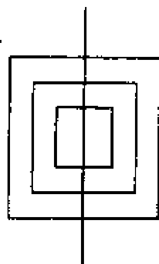
17. 何时可以画出一个城市里街道的各中心线而不重复经过一条街道（假设所有街道都是双向街道）？
18. 设计一个与算法 1 相似的过程，它在多重图里构造欧拉通路。

在练习 19～21 中，判定是否可以用一支铅笔连续移动、不离开纸面并且不重复地画出所示的图形。

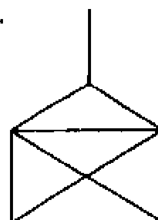
19.



20.



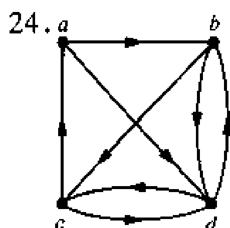
21.

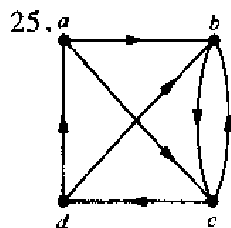


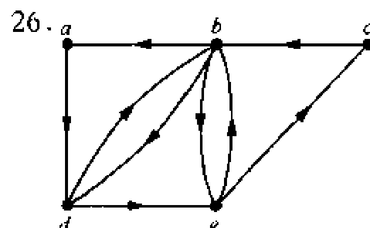
*22. 证明：不带有孤立点的有向多重图具有欧拉回路，当且仅当该图是弱连通的并且每个顶点的入度与出度都相等。

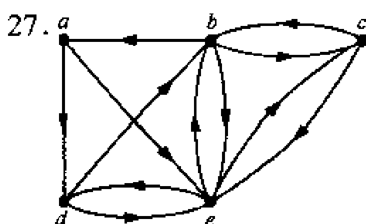
*23. 证明：不带有孤立点的有向多重图具有欧拉通路而没有欧拉回路，当且仅当该图是弱连通的并且存在两个顶点，一个顶点的入度比出度大 1 而另外一个顶点的出度比入度大 1，其余每个顶点的入度与出度都相等。

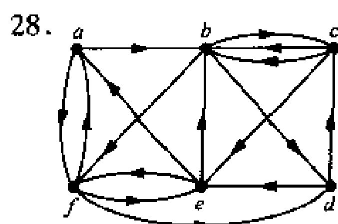
在练习 24～28 中，确定所示的有向图是否具有欧拉回路。若存在欧拉回路，则构造出一条欧拉回路。

24.


25.


26.


27.


28.


29. 确定练习 24 中的有向图是否具有欧拉通路。若存在欧拉通路，则构造出一条欧拉通路。

30. 确定练习 25 中的有向图是否具有欧拉通路。若存在欧拉通路，则构造出一条欧拉通路。

31. 确定练习 26 中的有向图是否具有欧拉通路。若存在欧拉通路，则构造出一条欧拉通路。

32. 确定练习 27 中的有向图是否具有欧拉通路。若存在欧拉通路，则构造出一条欧拉通路。

33. 确定练习 28 中的有向图是否具有欧拉通路。若存在欧拉通路，则构造出一条欧拉通路。

*34. 设计一个算法，用它构造有向图中的欧拉回路。

35. 设计一个算法，用它构造有向图中的欧拉通路。

36. 对哪些 n 值来说下列图具有欧拉回路？

　a) K_n　　b) C_n　　c) W_n　　d) Q_n

37. 对哪些 n 值来说练习 36 中的图具有欧拉通路而没有欧拉回路？

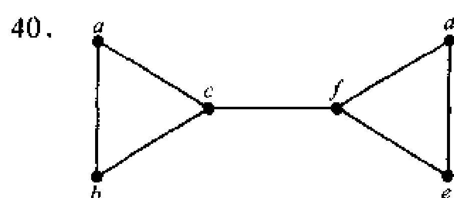
38. 对哪些 m 和 n 值来说完全偶图 $K_{m,n}$ 具有

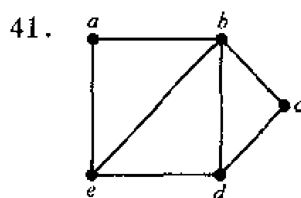
　a) 欧拉回路？

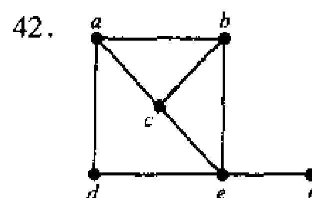
　b) 欧拉通路？

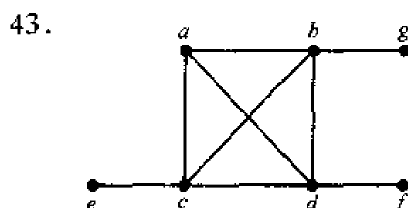
39. 当不重复任何部分地画出练习 1～7 中的每个图时，求出必须让铅笔离开纸面的最少次数。

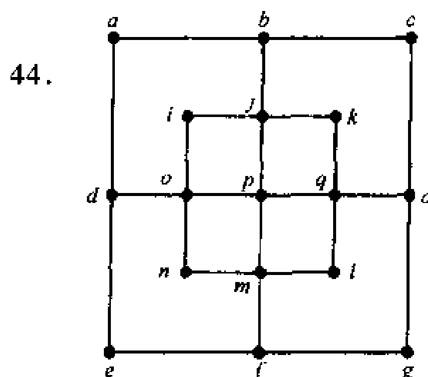
在练习 40～46 中，确定所给的图是否具有哈密顿回路。若有哈密顿回路，则求出这样一条回路。若没有哈密顿回路，则给出论证来说明这样的回路为什么不存在。

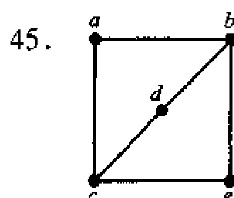
40. 

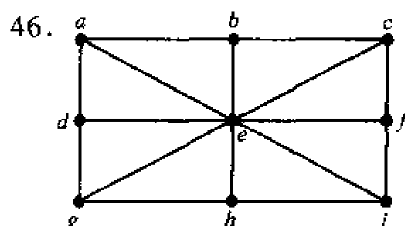
41. 

42. 

43. 

44. 

45. 

46. 

47. 练习 40 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

48. 练习 41 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

49. 练习 42 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

50. 练习 43 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

*51. 练习 44 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

52. 练习 45 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

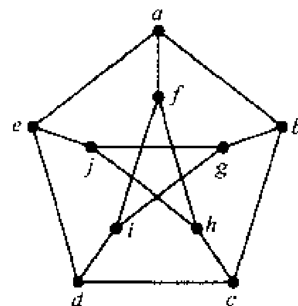
*53. 练习 46 中的图是否具有哈密顿通路？若有哈密顿通路，则求出这样一条通路。若没有哈密顿通路，则给出论证来说明这样的通路为什么不存在。

54. 对哪些 n 值来说练习 36 中的图具有哈密顿回路？

55. 对哪些 m 和 n 值来说完全偶图 $K_{m,n}$ 具有哈密顿回路？

*56. 证明：下图所示的彼得森图[⊖]没有哈密顿回路，但是删除顶点 v 和所有与 v 关联的边，所获得的子图却有哈密顿回路。

*57. 证明：每当 n 是正整数时，就存在 n 阶格雷码，或者等价地证明：$n>1$ 的 n 立方体 Q_n 总是具有哈密顿回路。[提示：用数学归纳法。证明如何从 $n-1$ 阶格雷码产生 n 阶格雷码。]



构造欧拉回路的弗留利算法是从连通多重图的任意一个顶点开始，连续地选择边来形成一条回路。一旦选择了一条边，就删除这条边。连续地选择边，使得每条边从上一条边结束的地方开始，而且使得这条边不是一条割边，除非别无选择。

58. 用弗留利算法找出在例 5 的图 G 中的欧拉回路。

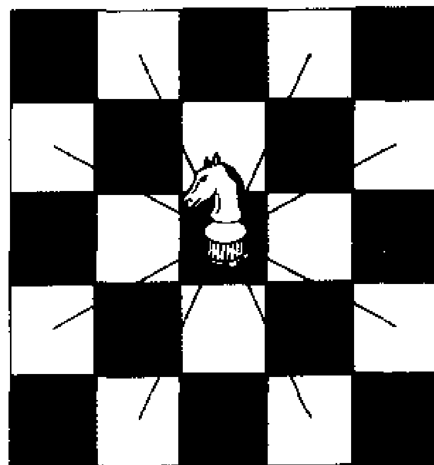
*59. 用伪代码表达弗留利算法。

**60. 证明：弗留利算法总是产生一条欧拉回路。

*61. 给出变化的弗留利算法来产生欧拉通路。

62. 一个诊断消息可以在计算机网络上发出，以便在所有连接和所有设备上执行测试。为了测试所有的连接，应当使用什么种类的通路？为了测试所有的设备呢？

63. 证明：带有奇数个顶点的偶图没有哈密顿回路。



国际象棋中的马，可以横向移动两格再纵向移动一格，或者横向移动一格再纵向移动两格。即在 (x, y) 格子的马可以移动到八个格子 $(x\pm2, y\pm1)$，$(x\pm1, y\pm2)$ 中的任何一个，只要这些格子在棋盘上，如右图所示。

⊖ 朱理乌斯·彼得·克里西安·彼得森（Julius Peter Christian Peterson，1839—1910）　彼得森出生在丹麦的索镇。其父是染匠。1854 年，他的父母再也负担不起他的学费，所以他到叔叔的杂货店当学徒。这位叔叔死的时候给彼得森留下足够的钱让他回到学校。毕业后他在哥本哈根工业学校开始学习工程，随后决定专攻数学。1858 年他出版了他的第一部教科书，一本关于对数的书。当他继承的遗产用完之后，他不得不靠教书来谋生。从 1859 年直到 1871 年，彼得森在哥本哈根的贵族私人高中教书。他一边教高中一边继续研究，在 1862 年进入哥本哈根大学。1862 年，他与劳拉·伯特森结婚；他们有三个孩子，两男一女。

1866 年，彼得森从哥本哈根大学获得数学学位，并且在 1871 年从该校最终获得博士学位。得到博士学位后，他在工业与军事高等专科学校任教。1887 年他被任命为哥本哈根大学的教授。在丹麦，彼得森作为一大系列的高中和大学教科书的作者而知名。其中他的一本《解决几何构造问题的方法和理论》被译成八种文字，英文版上次重印是在 1960 年而法文版最近在 1990 年重印，距离初版日期超过一个世纪。

彼得森研究的领域非常广泛，包括代数学、分析学、密码学、几何学、力学、数理经济学以及数论。他对图论的贡献，包括有关正则图的结果，是他最著名的工作。他以叙述的清晰性、解决问题的技巧性、独创性、幽默感、精力充沛以及善于教学而闻名。彼得森不愿意读其他数学家的著作。所以他经常重新发现别人已经证明过的结果，因此常常陷于尴尬之中。不过当其他数学家不读他的著作时他却常常发怒！

彼得森之死是哥本哈根报纸的头版新闻。当时一家报纸把他描述成科学界的汉斯·克里西安·安徒生——在学术世界里作出了贡献的人民之子。

马的周游是马从某个格子开始且访问每个格子恰好一次的合法移动的序列。若存在一种合法移动，它让马从周游的最后一个格子回到周游开始的地方，则马的周游称为*重返的*。可以用图为马的周游建模，棋盘上每个格子对应图中的一个顶点，若马可以在两个顶点所表示的格子之间合法地移动，则在图中用一条连接这两个顶点的边表示。

64. 画出表示马在 3×3 棋盘上合法移动的图。
65. 画出表示马在 3×4 棋盘上合法移动的图。
66. a) 证明：求马在 $m\times n$ 棋盘上的周游等价于求表示马在该棋盘上合法移动的图的哈密顿通路。
    b) 证明：求马在 $m\times n$ 棋盘上的重返的周游等价于求所对应的图上的哈密顿回路。
*67. 证明：存在马在 3×4 棋盘上的周游。
*68. 证明：不存在马在 3×3 棋盘上的周游。
*69. 证明：不存在马在 4×4 棋盘上的周游。
70. 证明：每当 m 和 n 都是正整数时，表示马在 $m\times n$ 棋盘上的合法移动的图就是偶图。
71. 证明：当 m 和 n 都是奇数时，不存在马在 $m\times n$ 棋盘上的重返的周游。[提示：利用练习 63、练习 66 b）和练习 70。]
*72. 证明：存在马在 8×8 棋盘上的周游。[提示：你可以用沃恩斯道夫（Warnsdorff）发明的下列方法来构造马的周游：从任意格子开始，然后总是移动到与最少数目的没有用过的格子连接的一个格子。虽然这个方法不能总是产生马的周游，但是它确实常常产生马的周游。]

7.6　最短通路问题

7.6.1　引言

许多问题可以用给边赋权值的图来建模。作为一个例子，考虑一下航线系统如何建模。用以下方法来建立基本的图模型：用顶点表示城市，用边表示航班。给边赋权值为城市之间的距离，就可以为涉及距离的问题建模。给边赋权值为飞行时间，就可以为涉及飞行时间的问题建模。给边赋权值为票价，就可以为涉及票价的问题建模。图 7-59 显示了给一个图的边的三种不同的赋值，分别表示距离、飞行时间和票价。

给每条边赋权值为一个数的图称为*带权图*。带权图用来为计算机网络建模。通信成本（比如租用电话线的月租费）、计算机在这些线路上的响应时间或计算机之间的距离等都可以用带权图来研究。图 7-60 显示了一些带权图，它们表示给计算机网络的图的边赋权值的三种方式，分别对应于成本、响应时间和距离。

与带权图有关的几种类型的问题频繁地出现。确定网络中两个顶点之间长度最短的通路就是一个这样的问题。说得更具体些，设带权图中一条通路的*长度*是这条通路上各边的权的总和。（读者应当注意，对*长度*一词的这种用法，与表示不带权的图中一条通路的边数的长度的用法是不同的。）问题是：什么是最短通路，即什么是在两个给定顶点之间长度最短的通路？例如，在图 7-59 所示带权图表示的航线系统里，在波士顿与洛杉矶之间以空中距离

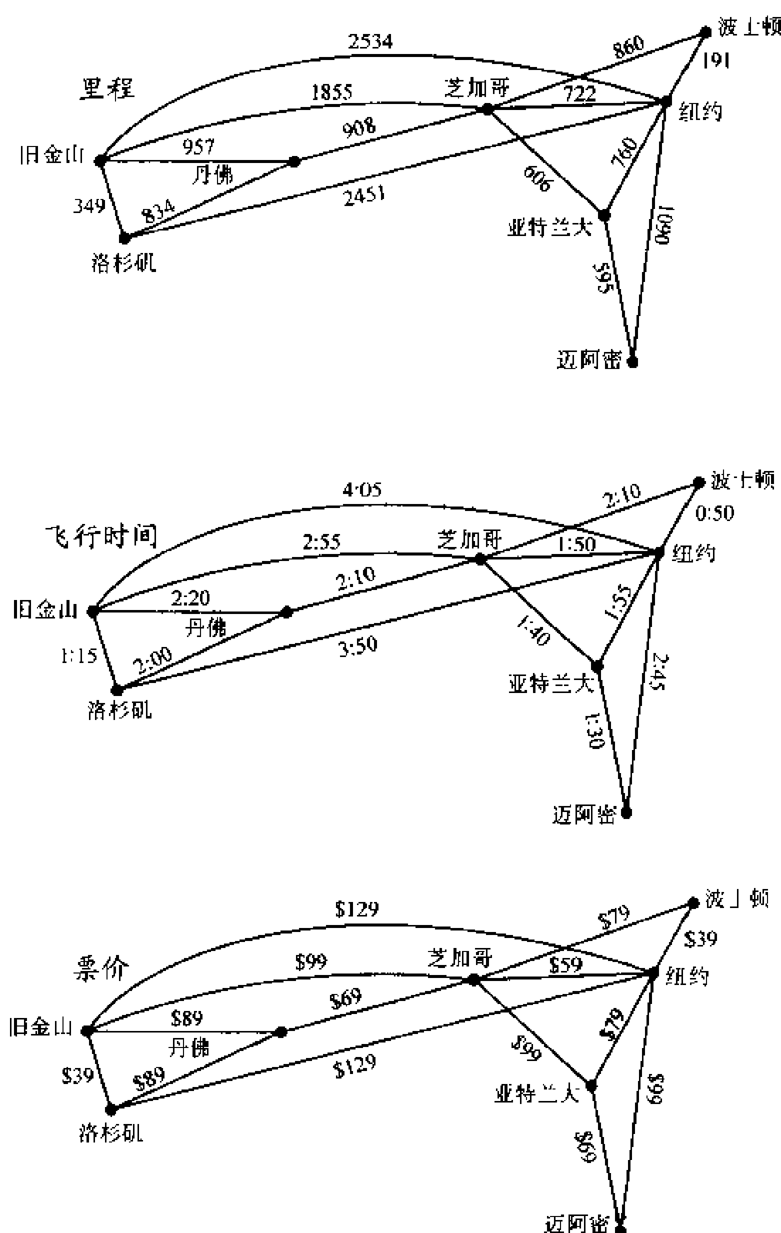


图 7-59　为航线系统建模的带权图

计算的最短通路是什么？在波士顿与洛杉矶之间什么样的航班组合的总飞行时间最短？（即在空中的总时间，不包括航班之间的时间）？在这两个城市之间的最低票价是什么？在图 7-60 所示的计算机网络中，连接旧金山的计算机与纽约的计算机所需要的最便宜的一组电话线是什么？哪一组电话线给出旧金山与纽约之间通信的最短响应时间？哪一组电话线有最短的总距离？

与带权图有关的另外一个重要问题是：求访问完全图每个顶点恰好一次的而总长度最短的回路。这就是著名的*旅行推销员问题*，它求一位推销员应当以什么样的顺序来访问其路程上的每个城市恰好一次，使得他旅行的总距离最短。将在本节后面讨论旅行推销员问题。

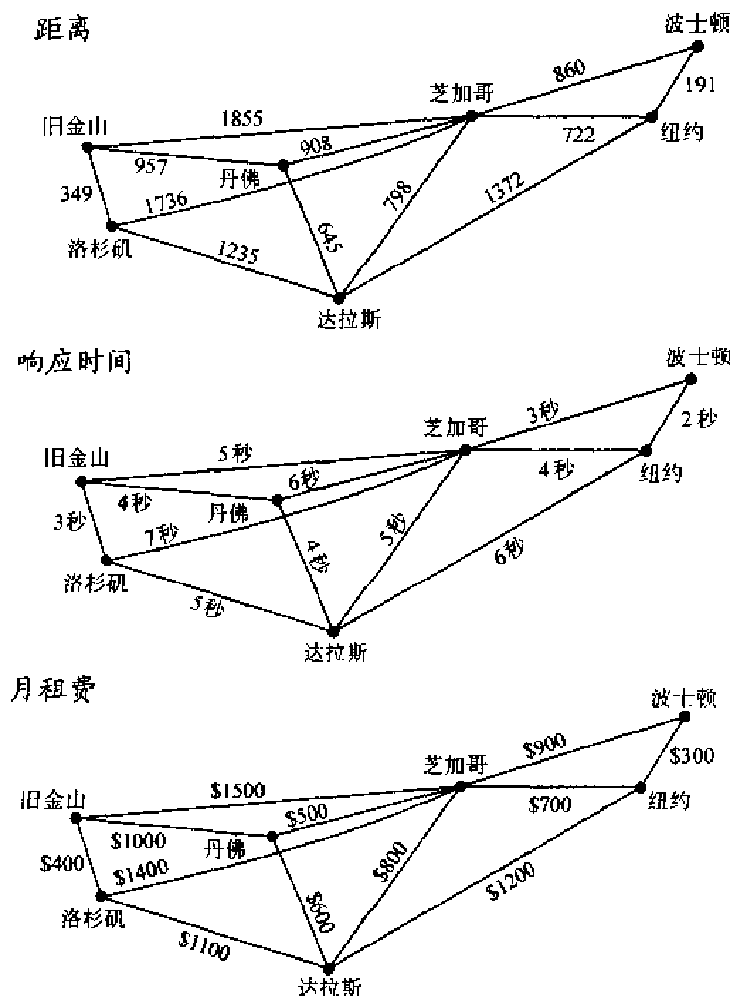


图 7-60　为计算机网络建模的带权图

7.6.2　一个最短通路算法

存在几个不同的算法来求带权图中两个顶点之间的最短通路。下面将给出荷兰数学家 E. 迪克斯屈拉[○]在 1959 年发现的一个算法。将要描述的算法，解决了无向带权图中的这个问题，其中所有的权都是正数。容易修改它来解决有向图中的最短通路问题。

在给出这个算法的形式化表示之前将给出一个启发性的例子。

例 1　在图 7-61 所示的带权图中，a 和 z 之间的最短通路的长度是什么？

○ 爱德思葛·韦伯·迪克斯屈拉（Edsger Wybe Dijkstra，生于 1930 年）　迪克斯屈拉出生在荷兰，20 世纪 50 年代早期，他在雷登大学学习理论物理时，就开始给计算机编程。在 1952 年，他意识到他对编程比对物理学更感兴趣，于是他迅速地结束了对物理学位的追求，转而开始了作为程序员的事业，即使当时编程还没有被认为是一种职业。（在 1957 年，阿姆斯特丹当局拒绝接受在他的结婚证上以“编程”作为他的职业。不过，当他把该条款改成“理论物理学家”时，他们却接受它。）

迪克斯屈拉一直是把编程作为一个科学学科的最有力的倡导者之一。他对诸多领域作出了奠基性的贡献，其中有操作系统（包括死锁避免），编程语言（包括结构化编程的概念），以及算法。在 1972 年迪克斯屈拉接受了美国计算机协会的图灵奖，这是计算机科学中最具声望的奖项之一。迪克斯屈拉在 1973 年成为宝来公司研究员，并在 1984 年被任命为位于奥斯汀的德克萨斯大学的计算机科学教授。

解　虽然通过检查就容易求出最短通路，但是我们要研究在理解迪克斯屈拉算法上有用的一些想法。将这样解决这个问题：求从 a 到各个相继的顶点的最短通路，直到到达 z 为止。

从 a 开始（直到到达终点为止）不包含除 a 以外的顶点的唯一通路是 a，b 和 a，d。因为 a，b 和 a，d 的长度分别为 4 和 2，所以 d 是与 a 最靠近的顶点。

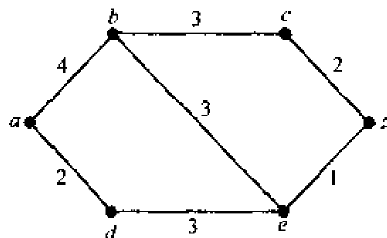


图 7-61　一个带权的简单图

可以通过查看（直到到达终点为止）只经过 a 和 d 的所有通路，来求出下一个最靠近 a 的顶点。到 b 的最短通路仍然是 a，b 长度为 4，而到 e 的最短通路是 a，d，e，长度为 5。所以，下一个与 a 最靠近的顶点是 b。

为了找出第三个与 a 最靠近的顶点，只需要检查（直到到达终点为止）只经过了 a，d 和 b 的那些通路。存在长度为 7 的到 c 的通路，即 a，b，c，以及长度为 6 的到 z 的通路，即 a，d，e，z。所以，z 是下一个与 a 最靠近的顶点，而且到 z 的最短通路的长度为 6。 ■

例 1 说明了在迪克斯屈拉算法中使用的一般原理。注意通过检查就可能求出从 a 到 z 的最短通路。不过，检查边数很多的图，无论对人还是对计算机来说都是不切实际的。

现在将考虑一般问题：在无向连通简单带权图中，求出 a 与 z 之间的最短通路的长度。迪克斯屈拉算法如下进行：求出从 a 到第一个顶点的最短通路的长度，从 a 到第二个顶点的最短通路的长度，依次类推，直到求出从 a 到 z 的最短通路的长度为止。

这个算法依赖于一系列的迭代。通过在每次迭代中添加一个顶点来构造出特殊顶点的集合。在每次迭代中完成一个标记过程。在这个标记过程中，用只包含特殊顶点的从 a 到 w 的最短通路的长度来标记 w。添加到特殊顶点集合中的顶点是在还没有成为特殊顶点的那些顶点中带有最小标记的那个顶点。

现在给出迪克斯屈拉算法的细节。它首先用 0 标记 a 而用∞标记其余的顶点。用记号 $L_0(a)=0$ 和 $L_0(v)=\infty$ 表示在没有发生任何迭代之前的这些标记（下标 0 表示“第 0 次”迭代）。这些标记是从 a 到这些顶点的最短通路的长度，其中这些通路只包含顶点 a。（因为不存在从 a 到除 a 外顶点的通路，所以∞是 a 与这样的顶点之间的最短通路的长度。）

迪克斯屈拉算法是通过形成特殊顶点的集合来进行的。设 S_k 表示在标记过程 k 次迭代之后的特殊顶点的集合。首先让 $S_0=\varnothing$。集合 S_k 是通过把不属于 S_{k-1} 的带最小标记的顶点 u 添加到 S_{k-1} 里来形成的。一旦把 u 添加到 S_k 中，就更新所有不属于 S_k 的顶点的标记，使得顶点 v 在第 k 个阶段的标记 $L_k(v)$ 是只包含 S_k 中顶点（即已有的特殊顶点集合再加上 u）的从 a 到 v 的最短通路的长度。

设 v 是不属于 S_k 的一个顶点。为了更新 v 的标记，注意 $L_k(v)$ 是只包含 S_k 中顶点的从 a 到 v 的最短通路的长度。当利用下面这个观察结果时，就可以有效地完成这个更新：从 a 到 v 的只包含 S_k 中顶点的最短通路，或者是从 a 到 v 的只包含 S_{k-1} 中顶点（即不包括 u 在内的特殊顶点）的最短通路，或者是在第 $k-1$ 阶段从 a 到 u 的最短通路加上边（u，v）。换句话说，即

$$L_k(a,v)=\min\{L_{k-1}(a,v),L_{k-1}(a,u)+w(u,v)\}$$

这个过程这样迭代：相继地添加顶点到特殊顶点集合里，直到添加 z 为止。当把 z 添加到特殊顶点集合里时，它的标记就是从 a 到 z 的最短通路的长度。在算法 1 里给出迪克斯屈拉算法。随后将给出这个算法为正确的证明。

算法 1　迪克斯屈拉算法

```
Procedure Dijkstra（G：所有权都为正数的带权连通简单图）
  {G 带有顶点 a = v0 , v1 , . . . , vn = z 和权 w(vi , vj)，其中若{vi , vj}不是 G 中的边，则
     w(vi , vj) = ∞}
  for i := 1 to n
       L(vi) := ∞
  L(a) := 0
  S := Ø
  {现在初始化标记，使得 a 的标记为 0 而所有其余标记为∞、而 S 是空集合}
  while z ∉ S
  begin
       u := 不属于 S 的L(u)最小的一个顶点
       S := S∪{u}
       for 所有不属于 S 的顶点 v
             if L(u) + w(u,v) < L(v) then L(v) := L(u) + w(u,v)
             {这样就给 S 中添加带最小标记的顶点并且更新不在 S 中的顶点的标记}
  end {L(z) = 从 a 到 z 的最短通路的长度}
```

下面的例子说明迪克斯屈拉算法是如何工作的。在这之后将证明这个算法总是产生带权图中两个顶点之间最短通路的长度。

例 2　用迪克斯屈拉算法求图 7-62a）所示的带权图中顶点 a 与 z 之间最短通路的长度。

解　在图 7-62 中显示迪克斯屈拉算法求 a 与 z 之间最短通路所用的步骤。在算法的每次迭代里，对集合 S_k 中的顶点加上圆圈。对每次迭代都标明了只包含 S_k 中顶点的从 a 到每个顶点的最短通路。当圆圈加到 z 时，算法终止。找到从 a 到 z 之间的最短通路是 a，c，b，d，e，z，长度为 13。■

注意　*在执行迪克斯屈拉算法的过程里，有时用一个表来代替对每步都重新画出这个图，这样更便于在每步里跟踪顶点的标记。*

下一步，用归纳论证来证明迪克斯屈拉算法产生无向连通带权图中两个顶点 a 与 z 之间最短通路的长度。用下列断言作为归纳假设：在第 k 次迭代中

(i) 在 S 中的顶点 v（$v \neq 0$）的标记是从 a 到这个顶点的最短通路的长度。

(ii) 不在 S 中的顶点的标记是（除了这个顶点自身之外）只包含 S 中顶点的从 a 到这个顶点的最短通路的长度。

当 $k=0$ 时，在没有执行任何迭代之前，$S=\{a\}$，所以从 a 到除 a 外的顶点的最短通路的长度是∞，而从 a 到 a 本身的最短长度为 0（在此允许一个通路不包含边在内）。因此基本情况是正确的。

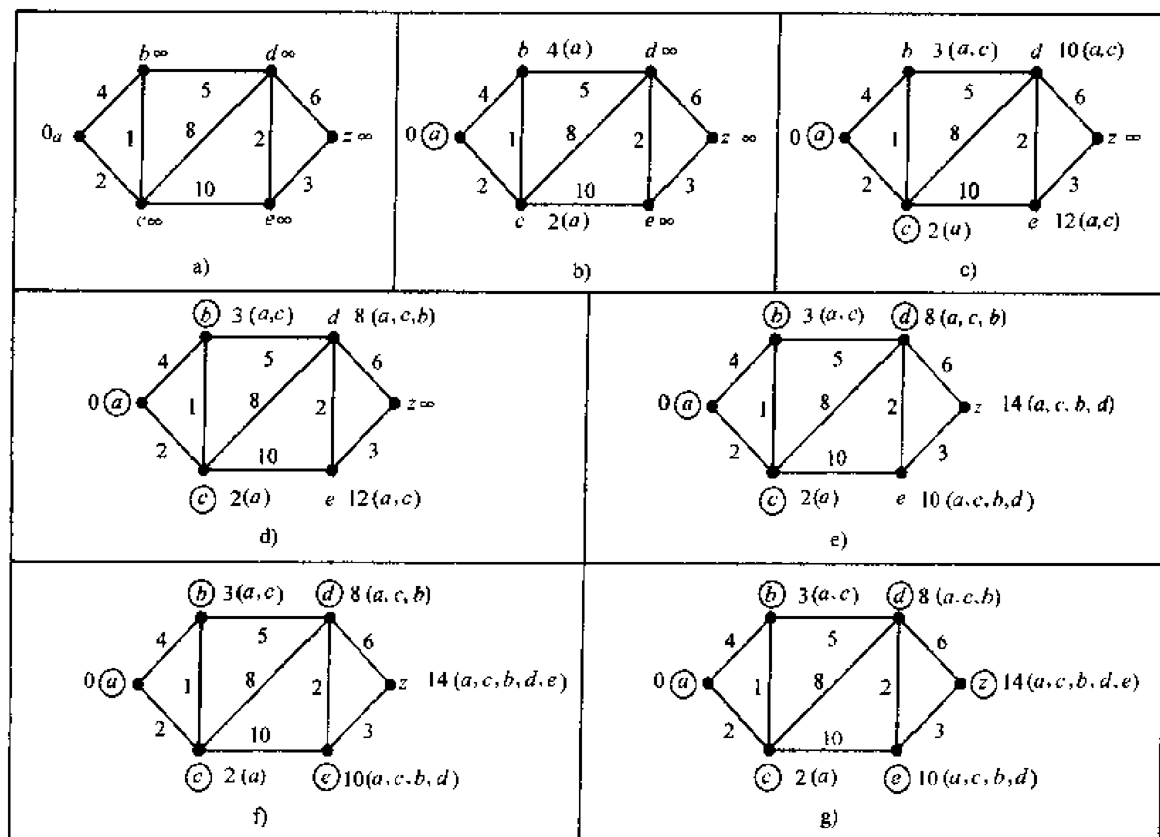


图 7-62　用迪克斯屈拉算法求从 a 到 z 的最短通路

假定归纳假设对第 k 次迭代成立。令 v 是在第 $k+1$ 次迭代中添加到 S 中的顶点，使得 v 是在第 k 次迭代结束时带最小标记的不在 S 中的顶点（在该顶点不唯一的情形里，可以采用带最小标记的任意顶点）。

根据归纳假设，可以看出在第 $k+1$ 次迭代之前，S 中的顶点都用从 a 出发的最短通路的长度来标记。另外，必须用从 a 到 v 的最短通路的长度来标记 v。假如情况不是这样，那么在第 k 次迭代结束时，就可能存在包含不在 S 中的顶点的长度小于$L_k(v)$的通路（因为 $L_k(v)$是在第 k 次迭代之后，只包含 S 中顶点的从 a 到 v 的最短通路的长度）。设 u 是在这样的通路里不属于 S 的第一个顶点。则存在一条从 a 到 u 的只包含 S 中顶点的长度小于$L_k(v)$的通路。这与对 v 的选择相矛盾。因此，在第 $k+1$ 次迭代结束时(i)成立。

设 u 是在第 $k+1$ 次迭代之后不属于 S 的一个顶点。从 a 到 u 的只包含 S 里顶点的最短通路要么包含 v，要么不包含 v。若它不包含 v，则根据归纳假设，它的长度是$L_k(u)$。若它确实包含 v，则它必然是这样组成的：一条从 a 到 v 的具有最短可能长度的通路，其中包含 S 中不同于 v 的元素，后面接着从 v 到 u 的边。在这种情形中它的长度是$L_k(v)+w(v,u)$。这样就证明了(ii)为真，因为$L_{k+1}(u)=\min\{L_k(u),L_k(v)+w(v,u)\}$。

已经证明了下面的定理。

定理 1　迪克斯屈拉算法求出连通简单无向带权图中两个顶点之间最短通路的长度。

现在可以估计迪克斯屈拉算法的计算复杂性（就加法和比较而言）。这个算法使用不超过 $n-1$ 次迭代，因为在每次迭代中添加一个顶点到特殊顶点集合里。若可以估计每次迭代

所使用的运算次数，则大功告成了。可以用不超过 $n-1$ 次比较来找出不在 S 中的带最小标记的顶点。然后，我们可以使用加法和比较来更新不在 S_k 中的每个顶点的标记。所以在每次迭代里使用不超过 $2(n-1)$ 次运算，因为在每次迭代里要更新的标记不超过 $n-1$ 个。因为使用不超过 $n-1$ 次迭代，每次迭代使用不超过 $2(n-1)$ 次运算，所以有下面的定理。

定理 2　迪克斯屈拉算法使用 $O(n^2)$ 次运算（加法和比较）来求出连通简单无向带权图中两个顶点之间最短通路的长度。

7.6.3 旅行推销员问题

现在讨论与带权图有关的一个重要问题。考虑下面的问题：一位旅行推销员想要访问 n 个城市中每个城市恰好一次，并且返回他的出发点。例如，假定这个推销员想要访问底特律、托雷多、萨吉挠、格兰特－拉皮兹以及卡拉玛祖（见图 7-63）。他应当以什么顺序访问这些城市以便旅行总距离最短？为了解决这个问题，可以假定旅行推销员从底特律出发（因为这个城市必须是回路的一部分），并且检查他访问其余四个城市然后返回底特律的所有可能方式（从别处出发将产生相同的回路）。存在总共 24 条这样的回路，但是因为当以相反顺序经过一条回路时，就经过了相同的距离，所以为了求出他必须旅行的最短总距离，只需要考虑 12 条不同的回路。列出这 12 条不同回路和每条回路旅行的最短总距离的表（表 7-4）。从这个表可以看出，使用底特律—托雷多—卡拉玛祖—格兰特－拉皮兹—萨吉挠—底特律的回路，就旅行了 458 英里的最短总距离。

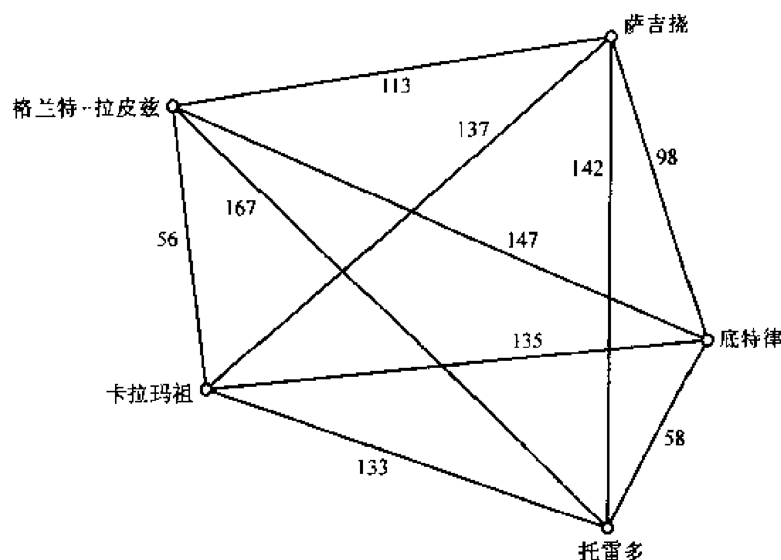


图 7-63　说明五个城市之间距离的图

只描述了旅行推销员问题的一个实例。旅行推销员问题求带权完全无向图中访问每个顶点恰好一次并且返回出发点的总权数最小的回路。这等价于求完全图中总权数最小的哈密顿回路，因为在回路中访问每个顶点恰好一次。

最直截了当的求解旅行推销员问题实例的方式是检查所有可能的哈密顿回路并且挑出总权数最小的一条回路。若在图中有 n 个城市，则为了求解这个问题，得检查多少条回路？一旦选定了出发点，需要检查的不同的哈密顿回路就有 $(n-1)!$ 条，因为第二个顶点有 $n-1$

表 7-4

路 线	总距离（英里）
底特律—托雷多—格兰特-拉皮兹—萨吉挠—卡拉玛祖—底特律	610
底特律—托雷多—格兰特-拉皮兹—卡拉玛祖—萨吉挠—底特律	516
底特律—托雷多—卡拉玛祖—萨吉挠—格兰特-拉皮兹—底特律	588
底特律—托雷多—卡拉玛祖—格兰特-拉皮兹—萨吉挠—底特律	458
底特律—托雷多—萨吉挠—卡拉玛祖—格兰特-拉皮兹—底特律	540
底特律—托雷多—萨吉挠—格兰特-拉皮兹—卡拉玛祖—底特律	504
底特律—萨吉挠—托雷多—格兰特-拉皮兹—卡拉玛祖—底特律	598
底特律—萨吉挠—托雷多—卡拉玛祖—格兰特-拉皮兹—底特律	576
底特律—萨吉挠—卡拉玛祖—托雷多—格兰特-拉皮兹—底特律	682
底特律—萨吉挠—格兰特-拉皮兹—托雷多—卡拉玛祖—底特律	646
底特律—格兰特-拉皮兹—萨吉挠—托雷多—卡拉玛祖—底特律	670
底特律—格兰特-拉皮兹—托雷多—萨吉挠—卡拉玛祖—底特律	728

种选择，第三个顶点有 $n-2$ 种选择，依次类推。因为可以用相反顺序来经过一条哈密顿回路，所以只需要检查 $(n-1)!/2$ 条回路来求出答案。注意 $(n-1)!/2$ 增长得极快。当只有几十个顶点时，试图用这种方式来解决旅行推销员问题就是不切实际的。例如，假如有 25 个顶点，那么就不得不考虑总共 $24!/2$（约为 3.1×10^{23}）条不同的哈密顿回路。假定检查每条哈密顿回路只花费 1 纳秒（10^{-9}秒），那么就需要大约 1 千万年才能求出这个图中长度最短的一条哈密顿回路。

因为旅行推销员问题同时具有实践和理论的重要性，所以已经投入了巨大的努力来设计解决它的有效算法。不过，还没有已知的解决这个问题的多项式最坏情形时间复杂性的算法。另外，假如找到了解决这个问题的多项式最坏情形时间复杂性的算法，那么许多其他困难问题（比如在第 1 章里讨论过的确定 n 个变元的命题公式是否重言式）也可以用多项式最坏情形时间复杂性算法来解决。这个结果是从 NP 完全性理论得出的。（关于这个理论的更多信息，参考本书末尾处关于这个题目的参考文献。）

当有许多需要访问的顶点时，解决旅行推销员问题的实际方法是使用近似算法。近似算法是这样的算法，它们不必产生问题的精确解，取而代之的是保证产生接近精确解的解。即它们可能产生带总权数 W' 的哈密顿回路，使得 $W\leqslant W'\leqslant cW$，其中 W 是精确解的总长度，而 c 是一个常数。例如，存在多项式最坏情形时间复杂性算法使得 $c=3/2$。在实际中，已经开发出这样的算法，它们可以只用几分钟的机时，就解决多达 1000 个顶点的旅行推销员问题，误差在精确解的 2% 之内。关于旅行推销员问题的更多信息，包括历史、应用和算法等，见《离散数学应用》（*Applications of Discrete Mathematics*）[MiRo91] 关于这个题目的那一章。

练习

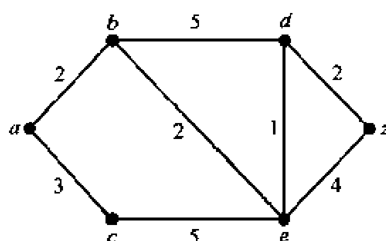
1. 对下列关于地铁系统的每个问题，描述一个可以用来解决这个问题的带权图模型。
   a) 在两站之间旅行所需要的最短时间是什么？

b）从一站到达另外一站所经过的最短距离是什么？

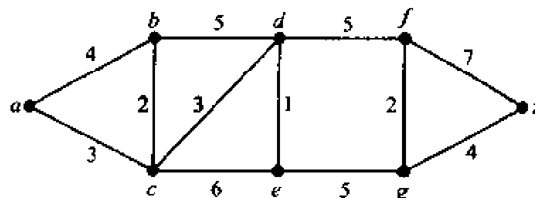
c）若把各站之间的票价求和就给出总票价，则在两站之间的最低票价是什么？

在练习 2～4 中，求给定带权图中 a 与 z 之间的最短通路的长度。

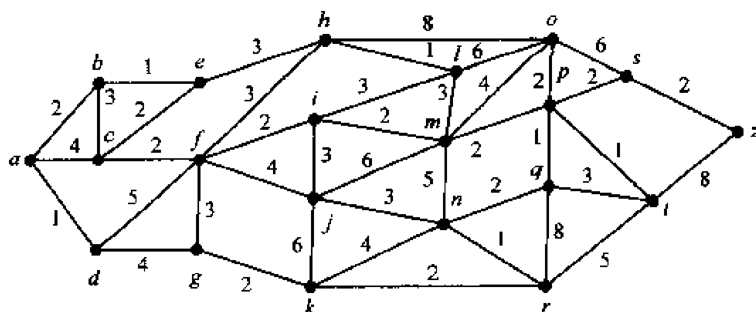
2.



3.



4.



5. 在练习 2～4 中的每个带权图中，在 a 与 z 之间的最短通路是什么？

6. 在练习 3 的带权图中，求下列成对顶点之间的最短通路的长度。

a）a 和 d　　b）a 和 f　　c）c 和 f　　d）b 和 z

7. 在练习 3 的带权图中，求练习 6 中的成对顶点之间的最短通路。

8. 在图 7-59 所示的航线系统中，求下列每对城市之间的最短通路（以英里表示）。

a）纽约与洛杉矶

b）波士顿与旧金山

c）迈阿密与丹佛

d）迈阿密与洛杉矶

9. 利用图 7-59 所示的飞行时间，求连接练习 8 中成对城市的总飞行时间最短的航班组合。

10. 利用图 7-59 所示的票价，求连接练习 8 中成对城市的成本最低的航班组合。

11. 在图 7-60 所示的通信网络里，求下列每对城市的计算机中心之间距离最短的路线。

a）波士顿与洛杉矶

b）纽约与旧金山

c）达拉斯与旧金山

d）丹佛与纽约

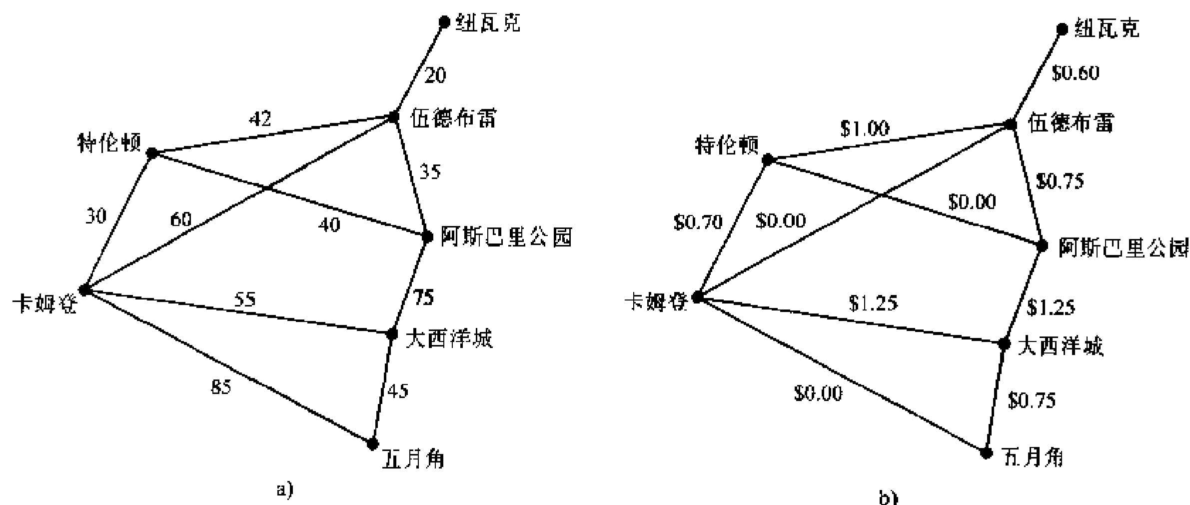
12. 利用在图 7-60 中给出的响应时间，求在练习 11 中成对的计算机中心之间响应时间最短的路线。

13. 利用在图 7-60 中给出的租费，求在练习 11 中成对的计算机中心之间月租费最便宜的路线。

14. 解释如何求无向图中两个顶点之间边数最少的通路，这里把它当作带权图里的最短通路问题。

15. 推广求带权简单有向连通图中两个顶点之间最短通路的迪克斯屈拉算法，以便求出在顶点 a 与图中其余每个顶点之间的最短通路的长度。

16. 推广求带权简单有向连通图中两个顶点之间最短通路的迪克斯屈拉算法，以便构造出在这些顶点之间的最短通路。
17. 在下图中的带权图说明了新泽西的一些主要道路。图中 a）说明了这些道路上的城市之间的距离；b）说明了通行费。



a）利用这些道路，求在纽瓦克与卡姆登之间以及在纽瓦克与五月角之间距离最短的路线。

b）利用图中在本题 a）中成对城市之间的道路，求就总通行费而言最便宜的路线。

18. 若各边的权都是不同的，则在带权图中两个顶点之间的最短通路是否唯一？
19. 若在应用中必须求出带权图中两个顶点之间的最长简单通路，这是一些什么样的应用？
20. 什么是图 7-62 的带权图中在 a 与 z 之间的最长简单通路的长度？在 c 与 z 之间呢？

可以用弗洛伊德算法来求出带权连通简单图中所有顶点对之间最短通路的长度。不过，不能用这个算法来构造出最短通路。(在下面，把无穷权值赋给任何一对不被图中的边所连接的顶点上。)

算法 2　弗洛伊德算法

procedure *Floyd*（G：带权简单图）
{如果 (v_i, v_j) 不是一条边，则 G 有顶点 $v_1, v_2, \ldots, v_n$ 和 $w(v_i, v_j) = \infty$ 的权 $w(v_i, v_j)$}
for $i := 1$ **to** n
　　for $j := 1$ **to** n
　　　　$d(v_i, v_j) := w(v_i, v_j)$
for $i := 1$ **to** n
　　for $j := 1$ **to** n
　　　　for $k := 1$ **to** n
　　　　　　if $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$ **then** $d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)$
{$d(v_i, v_j)$是在 v_i 和 v_j 之间最短路径的长度}

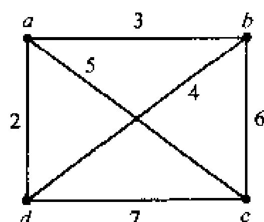
21. 用弗洛伊德算法求图 7-62 中带权图里所有顶点对之间的距离。

*22. 证明：弗洛伊德算法确定了带权简单图里所有顶点对之间的最短距离。

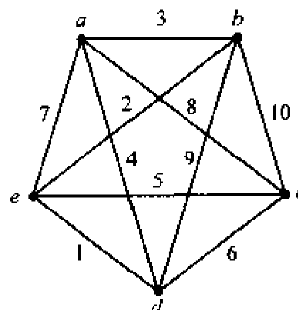
*23. 用弗洛伊德算法确定在带有 n 个顶点的带权简单图里所有顶点对之间的最短距离，给出算法运算（比较和加法）次数的大 O 估计。

*24. 证明：若边有负权，则迪克斯屈拉算法或许不能给出正确答案。

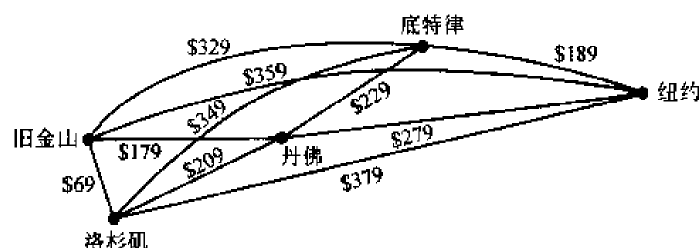
25. 通过求出所有哈密顿回路的总权数并且确定出总权数最小的回路，来解决下图的旅行推销员问题。



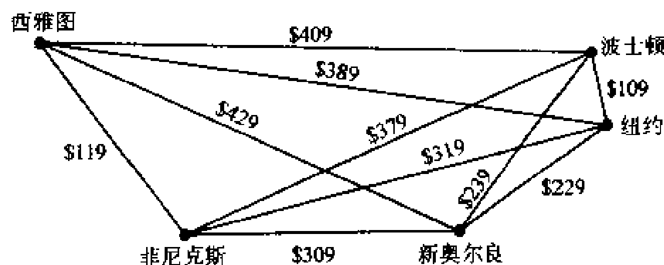
26. 通过求出所有哈密顿回路的总权数并且确定出总权数最小的回路，来解决下图的旅行推销员问题。



27. 求访问下图中每个城市的机票总价最低的路线，其中边上的权是在这两个城市之间的航班所提供的最低票价。



28. 求访问下图中每个城市的机票总价最低的路线，其中边上的权是在这两个城市之间的航班所提供的最低票价。



29. 构造一个带权无向图，使得对于访问某些顶点超过一次的回路来说，访问每个顶点至少一次的回路的总权数是最小的。[提示：存在带三个顶点的例子。]

30. 证明：求访问带权图每个顶点至少一次的总权数最小的回路的问题，可以归约为求访问带权图每个顶点恰好一次的总权数最小的回路的问题。这样做的方法是：构造一个新的带权图，它与原图有相同的顶点和边，但是连接顶点 u 和 v 的边的权却等于在原图中从 u 到 v 的通路的最小总权数。

7.7 平面性图

7.7.1 引言

考虑把三座房屋与三种设施每种都连接起来的问题，如图 7-64 所示。是否有可能这样来连接这些房屋与设施，使得在这样的连接里不发生交叉？这个问题可以用完全偶图 $K_{3,3}$来建模。原来的问题可以重新叙述为：能否在平面里画出 $K_{3,3}$，使得没有两条边发生交叉？

在本节里将研究能否在平面里画出一个让边不交叉的图的问题。特别是，将回答这个房屋与设施的问题。

总是存在许多方式来表示一个图。何时可能找出至少一种方式来在平面里表示这个图而使边没有任何交叉？

图 7-64　三座房屋与三种设施

定义　若可以在平面里画出一个图而让边没有任何交叉（其中边的交叉是表示边的直线或弧线在它们的公共端点以外的地方相交），则这个图是平面性的。这样一种画法称为这个图的平面表示。

即使通常带交叉地画出了一个图，这个图也仍然可能是平面性的，这是因为有可能以不同的方式来不带交叉地画出这个图。

例 1　K_4(图 7-65 所示，有两条边交叉)是否为平面性的？

解　K_4 是平面性的，因为可以不带交叉地画出它，如图 7-66 所示。■

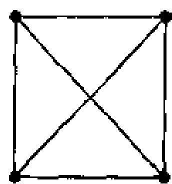


图 7-65　图 K_4

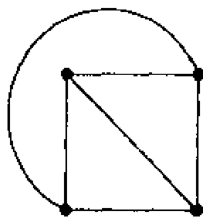


图 7-66　不带交叉地画出的 K_4

例 2　图 7-67 所示的 Q_3 是否为平面性的？

解　Q_3 是平面性的，因为可以画出它而没有任何边交叉，如图 7-68 所示。■

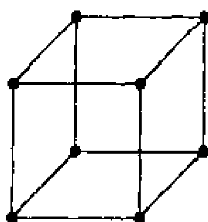


图 7-67　图 Q_3

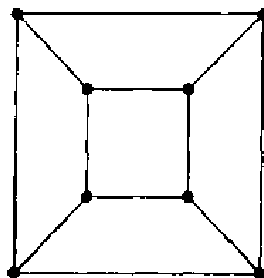


图 7-68　一种平面表示

可以通过显示一种平面表示来证明一个图是平面性的。更困难的是证明一个图是非平面性的。将给出一个例子说明如何以一种特别的方式来做到这一点。

例 3　图 7-69 所示的 $K_{3,3}$是否为平面性的？

解　在平面里画出 $K_{3,3}$而没有边交叉，任何这样的尝试都注定是失败的。现在说明这是为什么。在 $K_{3,3}$的任何平面表示里，顶点 v_1 和 v_2 都必须同时与 v_4 和 v_5 连接。这四条边所形成的封闭曲线把平面分割成两个区域 R_1 和 R_2，如图 7-70a）所示。顶点 v_3 属于 R_1 或 R_2。当 v_3 属于闭曲线的内部 R_2 时，在 v_3 和 v_4 之间以及在 v_3 和 v_5 之间的边，把 R_2 分割成两个区域 R_{21}和 R_{22}，如图 7-70b）所示。

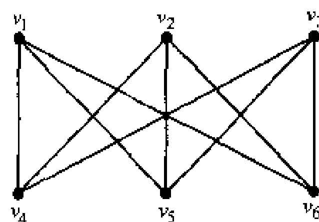


图 7-69　图 $K_{3,3}$

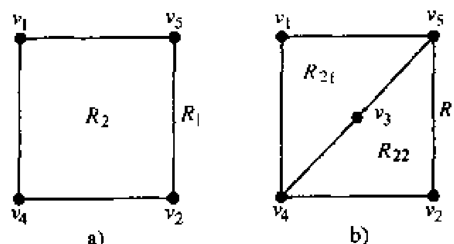


图 7-70　证明 $K_{3,3}$是非平面性的

下一步，注意没有办法来放置最后一个顶点 v_6 而又不迫使发生交叉。因为若 v_6 属于 R_1，则不能不带交叉地画出 v_6 和 v_3 之间的边。若 v_6 属于 R_{21}，则不能不带交叉地画出 v_2 和 v_6 之间的边。若 v_6 属于 R_{22}，则不能不带交叉地画出 v_1 和 v_6 之间的边。

当 v_3 属于 R_1 时，可以使用类似的论证。把完成这个论证留给读者（见本节末尾的练习 8）。所以 $K_{3,3}$是非平面性的。■

例 3 解决了在本节开头所描述的设施与房屋的问题。不能在平面里连接这三座房屋与三种设施而不发生交叉。可以用类似的论证来证明 K_5 是非平面性的。（见本节末尾的练习 9。）

7.7.2　欧拉公式

一个图的平面表示把平面分割成一些区域，包括一个无界的区域。例如，图 7-71 所示的图的平面表示把平面分割成六个区域。在该图中标记了这些区域。欧拉证明了一个图的所有平面表示都把平面分割成相同数目的区域。他是这样完成证明的：求出平面图的区域数、顶点数以及边数之间的关系。

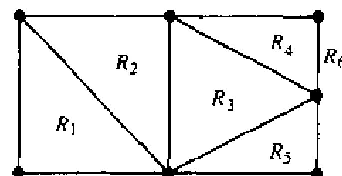


图 7-71　图的平面表示中的区域

定理 1　欧拉公式　设 G 是带 e 条边和 v 个顶点的连通平面性简单图。设 r 是 G 的平面表示里的区域数。则 $r=e-v+2$。

证　首先规定 G 的平面表示。将要这样证明定理：构造一系列子图 G_1，G_2，…，$G_e=G$，相继地在每个阶段上添加一条边。用下面的归纳定义来这样做。任意地选择一条 G 的边来获得 G_1。从 G_{n-1}这样获得 G_n：任意地添加一条与 G_{n-1}里顶点相关联的边，若与这条边关联的另一个顶点还不在 G_{n-1}里，则添加这个顶点。这样的构造是可能的，因为 G 是连通的。在添加 e 条边之后就获得 G。设 r_n，e_n 和 v_n 分别表示 G 的平面表示所诱导出的 G_n 的平面表示的区域数、边数和顶点数。

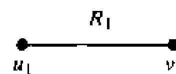


图 7-72　欧拉公式的证明中的基础情形

现在通过归纳来进行证明。对 G_1 来说关系 $r_1 = e_1 - v_1 + 2$ 为真，因为 $e_1 = 1$，$v_1 = 2$，而 $r_1 = 1$。这种情形如图 7-72 所示。

现在假定 $r_n = e_n - v_n + 2$。设 $\{a_{n+1}, b_{n+1}\}$ 是为了获得 G_{n+1} 而添加到 G_n 里的边。有两种情形需要考虑。在第一种情形里，a_{n+1} 和 b_{n+1} 都已经在 G_n 里了。这两个顶点必然是在一个公共区域 R 的边界上，否则就不可能把边 $\{a_{n+1}, bn+1\}$ 添加到 G_n 里而没有两条边交叉（并且 G_{n+1} 是平面性的。）这条新边的添加把 R 分割成两个区域。所以，在这种情形里，$r_{n+1} = r_n + 1$，$e_{n+1} = e_n + 1$，而且 $v_{n+1} = v_n$。因此，联系着区域数、边数、顶点数的公式两边都恰好增加一，所以这个公式仍然为真。换句话说，$r_{n+1} = e_{n+1} - v_{n+1} + 2$。在图 7-73 a）里说明这种情形。

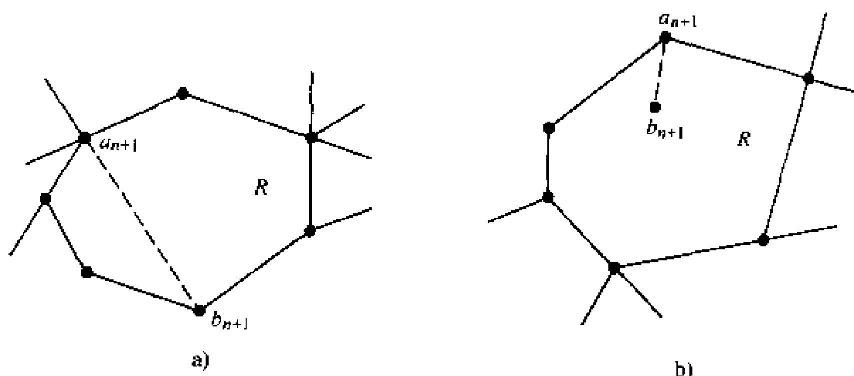


图 7-73　添加一条边到 G_n 来产生 G_{n+1}

在第二种情形里，新边的两个顶点之一已不在 G_n 里。假定 a_{n+1} 在 G_n 里但是 b_{n+1} 不在 G_n 里。添加这条新边不产生任何新的区域，因为 b_{n+1} 必然是在边界上有 a_{n+1} 的一个区域里。所以，$r_{n+1} = r_n$。另外，$e_{n+1} = e_n + 1$，而且 $v_{n+1} = v_n + 1$。联系着区域数、边数、顶点数的公式两边都保持相等，所以这个公式仍然为真。换句话说，$r_{n+1} = e_{n+1} - v_{n+1} + 2$。在图 7-73 b）里说明这种情形。

已经完成了归纳论证。因此对所有 n 来说都有 $r_n = e_n - v_n + 2$。因为原图是在添加了 e 条边之后所获得的图 G_e，所以这个定理为真。■

在下面的例子里说明欧拉公式。

例 4　假定连通平面性简单图有 20 个顶点，每个顶点的度都是 3。这个平面性图的平面表示把平面分割成多少个区域？

解　这个图有 20 个顶点，每个顶点的度为 3，所以 $v = 20$。因为这些顶点的度之和 $3v = 3\cdot 20 = 60$ 是等于边数的两倍 $2e$，所以有 $2e = 60$，或 $e = 30$。所以，根据欧拉公式，区域数是

$$r = e - v + 2 = 30 - 20 + 2 = 12$$ ■

可以用欧拉公式来建立平面性图所必须满足的一些不等式。在下面的推论里给出一个这样的不等式。

推论 1　若 G 是带 e 条边和 v 个顶点的连通平面性简单图，其中 $v \geqslant 3$，则 $e \leqslant 3v - 6$。

推论 1 的证明是基于区域的度数的概念，它定义为这个区域边界上的边数。当一条边在边界上出现两次（所以当描画边界时就描画它两次）时，它给度数贡献 2。图 7-74 里所示的图的区域的度数都显示在图中。

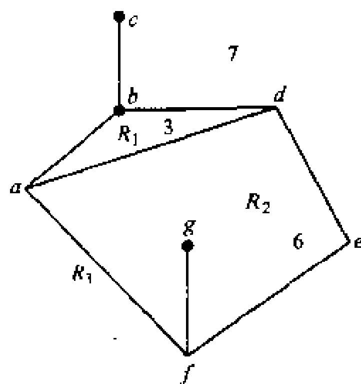


图 7-74　区域的度数

现在可以给出推论 1 的证明了。

证　画在平面里的连通平面性简单图把平面分割成区域，比如说 r 个区域。每个区域的度数至少为 3。（因为这里所讨论的图都是简单图，不允许带有可能产生 2 度区域的多重边，或者可能产生 1 度区域的环。）特别是，注意无界区域的度数至少为 3，因为在图里至少有 3 个顶点。

注意各区域的度数之和恰好是图中边数的两倍，因为每条边都在区域的边界上出现两次（或者在两个不同区域里，或者两次在相同的区域里）。因为每个区域都有大于或等于 3 的度数，所以

$$2e=\sum_{\text{所有区域}R}\deg(R)\geqslant 3r$$

因此，

$$(2/3)e\geqslant r$$

利用 $r=e-v+2$（欧拉公式），就得到

$$e-v+2\leqslant(2/3)e$$

所以得到 $e/3\leqslant v-2$。这样就证明了 $e\leqslant 3v-6$。■

可以用这个推论来证明 K_5 是非平面性的。

例 5　用推论 1 证明：K_5 是非平面性的。

解　图 K_5 有 5 个顶点和 10 条边。不过，对这个图来说，不满足不等式 $e\leqslant 3v-6$，因为 $e=10$ 和 $3v-6=9$。因此，K_5 不是平面性的。■

前面已经证明了 $K_{3,3}$不是平面性的。不过，注意这个图有 6 个顶点和 9 条边。这意味着满足不等式 $e=9\leqslant 12=3\cdot 6-6$。所以，满足不等式 $e\leqslant 3v-6$ 的事实并不蕴涵着一个图是平面性的。不过，可以利用定理 1 的下面的推论来证明 $K_{3,3}$不是平面性的。

推论 2　若连通平面性简单图有 e 条边和 v 个顶点，$v\geqslant 3$ 并且没有长度为 3 的回路，则 $e\leqslant 2v-4$。

推论 2 的证明类似于推论 1 的证明，不同之处在于，在这种情形里，没有长度为 3 的回路的事实蕴函着区域的度数必然至少为 4。把这个证明的细节留给读者（见本节末尾的练习 13）。

例 6　用推论 2 来证明 $K_{3,3}$是非平面性的。

解　因为 $K_{3,3}$没有长度为 3 的回路（容易看出这一点，因为它是偶图），所以可以使用推论 2。$K_{3,3}$有 6 个顶点和 9 条边。因为 $e=9$ 和 $2v-4=8$，所以推论 2 证明 $K_{3,3}$是非平面性的。■

7.7.3 库拉图斯基定理

已经看到 $K_{3,3}$ 和 K_5 都不是平面性的。显然，若一个图包含这两个图作为子图，则它不是平面性的。另外，所有非平面性的图必然包含一个子图，它是可以利用某些允许的操作从 $K_{3,3}$ 或 K_5 来获得的。

若一个图是平面性的，则通过删除一条边 $\{u, v\}$ 并且添加一个新顶点 w 和两条边 $\{u, w\}$ 与 $\{w, v\}$，所获得的任何图也是平面性的。这样的操作称为**初等细分**。若可以从相同的图通过一系列初等细分来获得图 $G_1=(V_1,E_1)$ 和图 $G_2=(V_2,E_2)$，则它们称为是**同胚的**。在图 7-75 中所示的三个图都是同胚的，因为它们都可以从第一个图通过初等细分来获得。（读者应当确定出从 G_1 获得 G_2 和 G_3 所需要的初等细分的序列。）

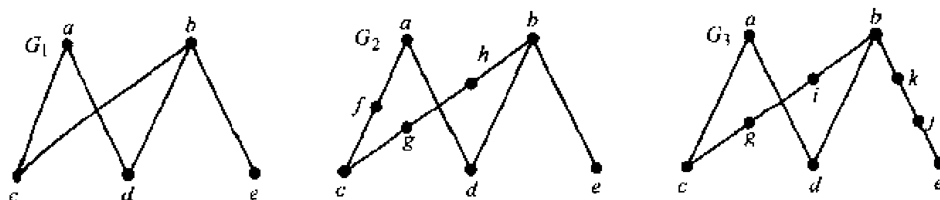


图 7-75　同胚的图

波兰数学家库拉图斯基[⊖]在 1930 年证明了下面的定理，这个定理用图同胚的概念刻画了平面性图。

定理 2　一个图是非平面性的，当且仅当它包含一个同胚于 $K_{3,3}$ 或 K_5 的子图。

显然，一个包含着同胚于 $K_{3,3}$ 或 K_5 的子图的图是非平面性的。不过，相反的命题——即每个非平面性图都包含一个同胚于 $K_{3,3}$ 或 K_5 的子图——的证明是复杂的，因而不在这里给出。下面的例子说明如何使用库拉图斯基定理。

例 7　确定图 7-76 中所示的图是否为平面性的。

解　G 有同胚于 K_5 的子图 H。H 是这样获得的：删除 h，j 和 k 以及所有与这些顶点关联的边。H 是同胚于 K_5 的，因为从 K_5（带有顶点 a，b，c，g 和 i）通过一系列初等细分，添加顶点 d，e 和 f 就可以获得 H。（读者应当构造出这样的一系列初等细分。）因此，G 是非平面性的。■

例 8　在图 7-77 a）中所示的彼得森图是否为平面性的？（丹麦数学家朱利乌斯·彼得森在

⊖ 卡兹米尔兹·库拉图斯基（Kazimierz Kuratowski，1896—1980）　库拉图斯基是华沙一位著名律师的儿子，他在华沙上了高中。他从 1913 年到 1914 年在苏格兰的格拉斯哥学习，但是在第一次世界大战爆发后无法返回那里。1915 年他进入华沙大学，在华沙大学他积极参加波兰的爱国学生运动。他在 1919 年发表了第一篇论文，并且在 1921 年获得博士学位。他是以华沙数学学派著称的小组里的活沃成员，从事集合论基础和拓扑学领域的工作。他被任命为勒沃工业大学的副教授，在那里呆了 7 年，与重要的波兰数学家巴拿赫和乌拉姆合作。在 1930 年还在勒沃时，库拉图斯基完成了刻画平面性图的工作。

在 1934 年他作为正教授回到华沙大学，直到第二次世界大战开始，他都活跃在研究和教学领域。在战争期间，因为对受过教育的波兰人的迫害，库拉图斯基隐姓埋名，并且在地下的华沙大学教书。在战后他帮助复兴波兰的数学，担任波兰国家数学研究所所长。他写过超过 180 篇的论文和三本被广泛使用的教科书。

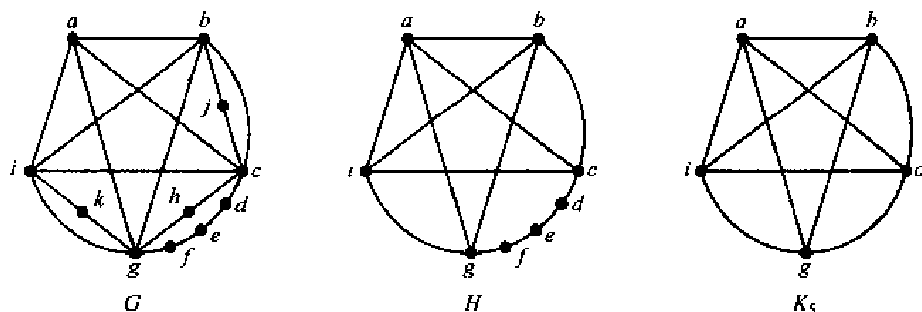


图 7-76　无向图 G、同胚于 K_5 的子图 H 和 K_5

1891 年介绍过这个图；常常用它来说明图的各种理论性质。)

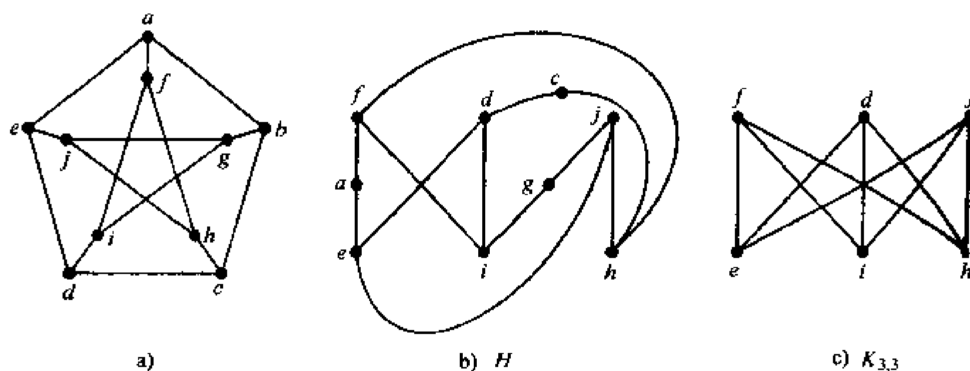


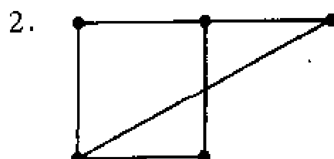
图 7-77　a) 彼得森图；b) 同胚于 $K_{3,3}$ 的子图 H；c) $K_{3,3}$

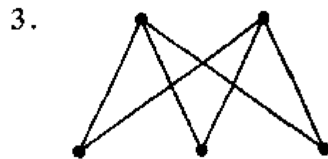
解　彼得森图的子图 H 是这样获得的：删除 b 和以 b 作为端点的三条边，如图 7-77 b) 所示，它是同胚于带有顶点集合 $\{f, d, j\}$ 和 $\{e, i, h\}$ 的 $K_{3,3}$ 的，这是因为可以通过一系列初等细分来获得它，包括删除 $\{d, h\}$ 并且添加 $\{c, h\}$ 和 $\{c, d\}$，删除 $\{e, f\}$ 并且添加 $\{a, e\}$ 和 $\{a, f\}$，删除 $\{i, j\}$ 并且添加 $\{g, i\}$ 和 $\{g, j\}$。因此，彼得森图不是平面性的。■

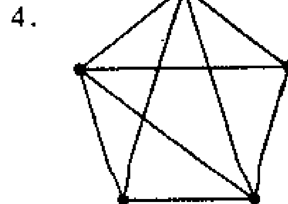
练习

1. 五座房屋能否不带连接交叉地与两种设施相连接？

在练习 2～4 中，不带任何交叉地画出给定的平面性图。

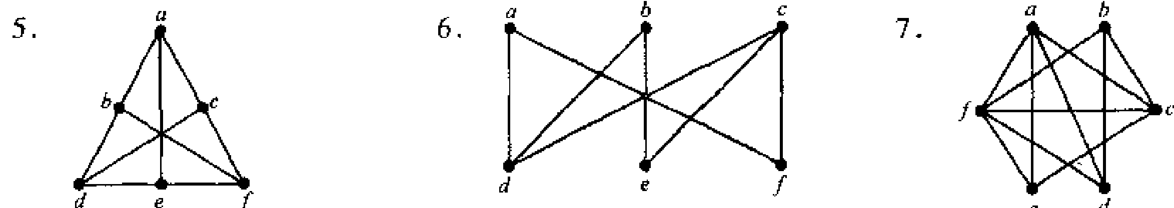
2. 

3. 

4. 

在练习 5～7 中，确定所给的图是否为平面性的。若是平面性的，则画出它使得没有边交叉。

5. 6. 7.



8. 完成在例 3 中的论证。

9. 用类似于在例 3 中给出的论证来证明：K_5 是非平面性的。

10. 假定一个连通平面性图有 8 个顶点，每个顶点的度都为 3。这个图的平面表示把平面分割成多少个区域？

11. 假定一个连通平面性图有 6 个顶点，每个顶点的度都为 4。这个图的平面表示把平面分割成多少个区域？

12. 假定一个连通平面性图有 30 条边。若这个图的平面表示把平面分割成 20 个区域，则这个图有多少个顶点？

13. 证明推论 2。

14. 假定一个连通的偶平面性图有 e 条边和 v 个顶点。证明：若 $v \geq 3$，则 $e \leq 2v-4$。

*15. 假定一个带有 e 条边和 v 个顶点的连通平面性简单图不包含长度为 4 或更短的回路。证明：若 $v \geq 4$，则$e \leq (5/3)v-(10/3)$。

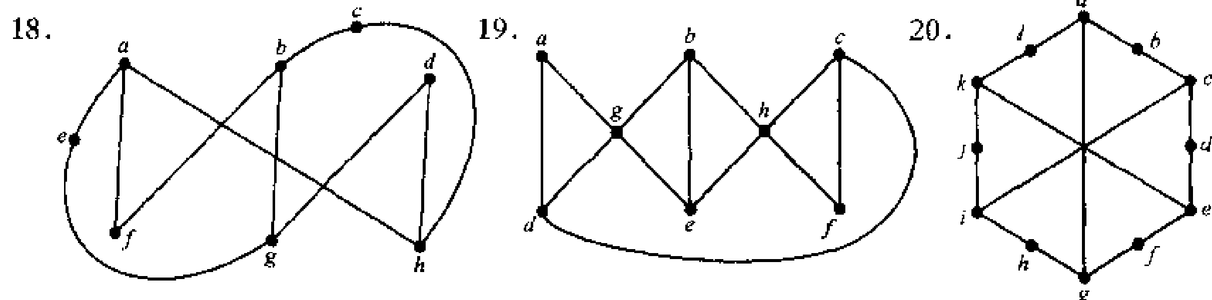
16. 假定一个平面性图有 k 个连通分支、e 条边和 v 个顶点。另外假定这个图的平面表示把平面分割成 r 个区域。求用 e，v 和 k 所表示的 r 的公式。

17. 下面的哪些非平面性图具有这样的性质：删除任何一个顶点以及与这个顶点关联的所有边就产生一个平面性图？

a) K_5　　b) K_6　　c) $K_{3,3}$　　d) $K_{3,4}$

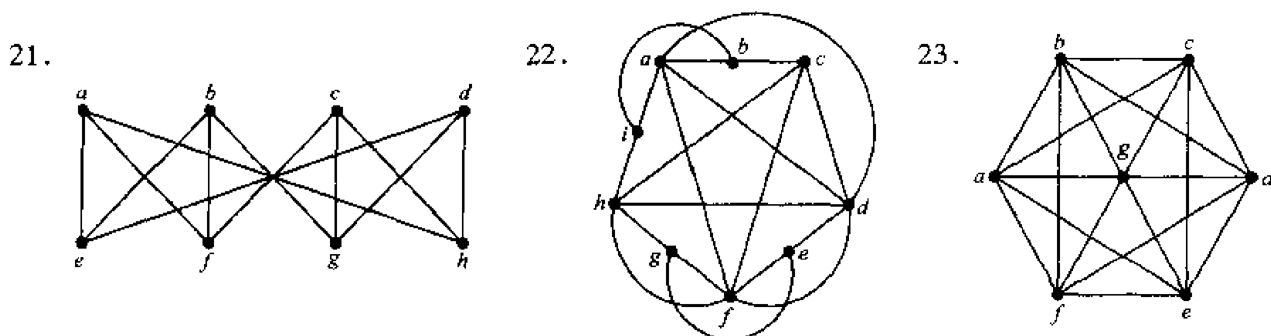
在练习 18～20 里，确定给定的图是否同胚于 $K_{3,3}$。

18. 19. 20.



在练习 21～23 中，用库拉图斯基定理来确定所给的图是否为平面性的。

21. 22. 23.




一个简单图的交叉数是指，当在平面里画出这个图，其中不允许任何三条表示边的弧线在同一个点交叉时，交叉的最少次数。

24. 证明：$K_{3,3}$的交叉数为1。

**25. 求下面每个非平面性图的交叉数。

a) K_5 b) K_6 c) K_7 d) $K_{3,4}$ e) $K_{4,4}$ f) $K_{5,5}$

*26. 求彼得森图的交叉数。

**27. 证明：若m和n都是偶正整数，则$K_{m,n}$的交叉数小于或等于$mn(m-2)(n-2)/16$。[提示：沿着x轴放置m个顶点，使得它们间距相等并且关于原点对称，再沿着y轴放置n个顶点，使得它们间距相等并且关于原点对称。现在连接x轴上m个顶点中的每一个与y轴上n个顶点中的每一个，并且计算交叉数。]

简单图G的厚度是指G的平面性子图的最小个数，这些子图以G作为它们的并集。

28. 证明：$K_{3,3}$的厚度为2。

*29. 求练习25中的图的厚度。

30. 证明：若G是一个带有v个顶点和e条边的连通简单图，则G的厚度至少为$\lceil e/(3v-6)\rceil$。

*31. 利用练习30来证明：每当n是正整数时，K_n的厚度就至少为$\lfloor (n+7)/6\rfloor$。

32. 证明：若G是一个带有v个顶点和e条边并且没有长度为三的回路的连通简单图，则G的厚度至少为$\lceil e/(2v-4)\rceil$。

33. 利用练习32来证明：每当m和n都是正整数时，$K_{m,n}$的厚度就至少是$\lceil mn/(2m+2n-4)\rceil$。

*34. 在一个torus（炸圈饼形状的固体）的表面上画出K_5，使得没有边交叉。

*35. 在一个torus的表面上画出$K_{3,3}$，使得没有边交叉。

7.8 图着色

7.8.1 引言


与区域地图（比如世界各部分的地图）着色有关的问题，已经在图论里产生了许多结果。当一幅地图着色时[⊖]，具有公共边界的两个区域在传统上指定为不同的颜色。确保两上相邻的区域永远没有相同的颜色，一种方法是对每个区域都使用不同的颜色。不过，这是低效的方法，而且在具有许多区域的地图上，可能难以区分相似的颜色。替代的方法是，应当尽可能地使用少数几种颜色。考虑这样的问题：确定可以用来给一幅地图着色的颜色的最小数目，使得相邻的区域永远没有相同的颜色。例如，对图7-78左侧所示的地图来说，四色是足够的，但是三色就不够。（读者应当验证这一点。）在图7-78右侧所示的地图里，三色是足够的（但是二色就不够）。

平面里的每幅地图都可以表示成一个图。为了建立这样的对应关系，地图的每个区域都表示成一个顶点。若两个顶点所表示的区域具有公共边界，则用边连接这两个顶点。只相交

⊖ 我们假定地图里所有区域都是连通的。这样就消除了像密歇根这样的地理实体所引起的任何问题。

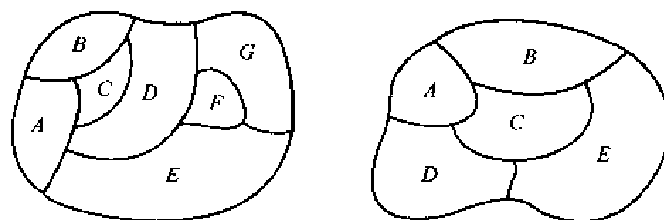


图 7-78 两幅地图

于一个点的两个区域不算是相邻的。这样所得到的图称为这个地图的对偶图。根据地图的对偶图的构造方式，显然在平面里的任何地图都具有平面性的对偶图。图 7-79 显示对应于图 7-78 中所示地图的对偶图。

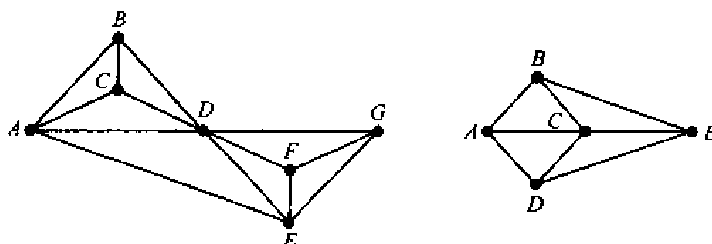


图 7-79 图 7-78 中地图的对偶图

给地图的区域着色的问题等价于这样的问题：给对偶图的顶点着色，使得在对偶图里没有两个相邻的顶点具有相同的颜色。给出下面的定义。

定义 1 简单图的着色是对该图的每个顶点都指定一种颜色，使得没有两个相邻的顶点指定为相同的颜色。

通过对每个顶点都指定一种不同的颜色，就可以着色一个图。不过，对大多数的图来说，可以找到所用颜色数少于图中顶点数的着色。那么，什么是所需要的最少颜色数？

定义 2 图的色数是着色这个图所需要的最少颜色数。

注意，求平面图的色数等于是求平面地图着色所需要的最少颜色数，使得没有两个相邻的区域指定为相同的颜色。这个问题已经研究了超过 100 年[⊖]。数学里最著名的定理之一提供了它的答案。

定理 1 *四色定理* 平面图的色数不超过 4。

四色定理最早是作为猜想在 19 世纪 50 年代提出的。美国数学家肯尼思·阿佩尔和沃尔夫冈·黑肯最终在 1976 年证明了它。在 1976 年之前，发表过许多不正确的证明，其中的错误常常难以发现。另外，还在通过画出需要超过四色的地图来构造反例上面做过许多无效的尝试。

⊖ 历史注记：在 1852 年，德摩根从前的一个学生弗郎西斯·古特利注意到，用四种颜色可以给英格兰的郡着色，使得没有相邻的郡被指定为相同的颜色。在这个证据的基础上，他猜想四色定理为真。弗郎西斯把这个问题告诉他的弟弟弗雷德里克，弗雷德里克当时是德摩根的学生。弗雷德里克就哥哥的猜想询问了他的老师德摩根。德摩根对这个问题极其感兴趣并且向数学界公布了它。事实上，在德摩根给威廉·罗万·哈密顿爵士的信中第一次在书面上提到这个猜想。虽然德摩根认为哈密顿可能对这个问题感兴趣，但是哈密顿却明显地对它不感兴趣，因为它与四元数毫无关系。

也许迄今为止在数学里最有名的错误证明就是伦敦的律师和业余数学家艾尔弗雷德·肯普[⊖]在 1879 年所发表的四色定理证明。数学家们一直把他的证明当做正确的证明来接受，直到 1890 年珀西·希伍德发现了一处错误，这个错误使得肯普的论证是不完全的。不过，事实证明，肯普的推理路线是阿佩尔和黑肯所给出的成功证明的基础。他们的证明依赖于计算机所完成的逐个对各种情形的仔细分析。他们证明，若四色定理为假，则存在一个反例，它是大约 2000 种不同类型中的一种，然后他们证明这些类型都没有导致反例。在他们的证明中使用了超过 1000h 的机时。这个证明引起了广泛争论，原因是计算机在里面起到如此重要的作用[⊖]。例如，在计算机程序里有没有导致不正确结果的错误？假如他们的论证依赖于或许不可靠的计算机输出，那么它是不是真正的证明？

四色定理只适用于平面图。在例 2 里将证明，非平面性图可以有任意大的色数。

证明一个图的色数为 n 需要做两件事。首先必须证明：用 n 种颜色可以着色这个图。构造出这样的着色就可以完成这件事。其次证明：用少于 n 种颜色不能着色这个图。下面的例子说明如何求出色数。

例 1 在图 7-80 中所示的图 G 和 H 的色数是多少？

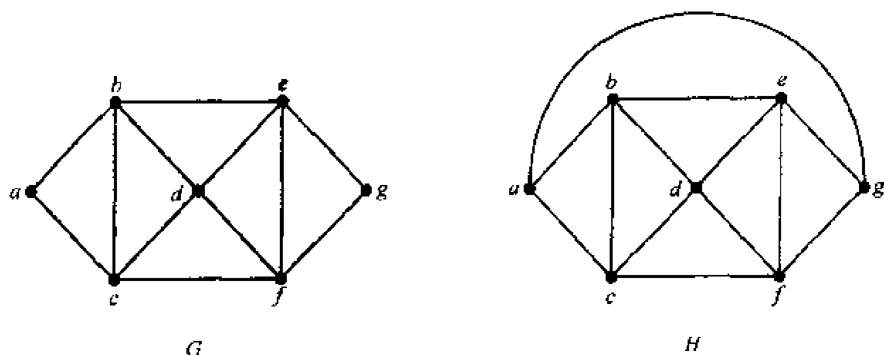


图 7-80 简单图 G 和 H

解 图 G 的色数至少为 3，因为顶点 a，b 和 c 必须指定为不同的颜色。为了看出是否可以用三种颜色来对 G 着色，指定红色给 a，蓝色给 b，以及绿色给 c。于是，可以（而且必须）把 d 着色成红色，因为它与 b 和 c 相邻。另外，可以（而且必须）把 e 着色成绿色，因为它只与着色成红色和蓝色的顶点相邻，同时可以（而且必须）把 f 着色成蓝色，因为它只与着色成红色和绿色的顶点相邻。最后，可以（而且必须）把 g 着色成红色，因为它只与着色成蓝色和绿色的顶点相邻。这样就产生出恰好使用三种颜色的 G 的着色。图 7-81 显示这样的着色。

图 H 是由图 G 和连接 a 与 g 的一条边所组成的。用三种颜色来着色 H 的任何尝试都必须遵循与着色 G 所用的同样的推理，不同之处是在最后阶段当除了 g 以外的所有顶点都已

⊖ 艾尔弗雷德·布雷·肯普（Alfred Bray Kempe，1849—1922） 肯普是一位律师和教会法规的主要权威。不过，在剑桥大学学习过数学之后，他保持了对数学的兴趣，并且在以后的生活中对数学研究投入了相当多的时间。肯普对运动学和数理逻辑做出过贡献，运动学是处理运动的数学分支。不过，肯普被人记住，主要还是因为他对四色定理的错误证明。

⊖ 历史注记：虽然罗布森、桑得尔斯、西摹尔以及托马斯在 1996 年找到了四色定理的简化证明，把证明的计算部分减少到检查 633 种格局，但是仍然还没有找到不依赖于大量计算的证明。

经着色时。那时，因为 g 与着色成红色、蓝色和绿色的顶点（在 H 里）相邻，所以需要使用第四种颜色，比如褐色。因此，H 的色数为 4。在图 7-81 显示 H 的一种着色。■

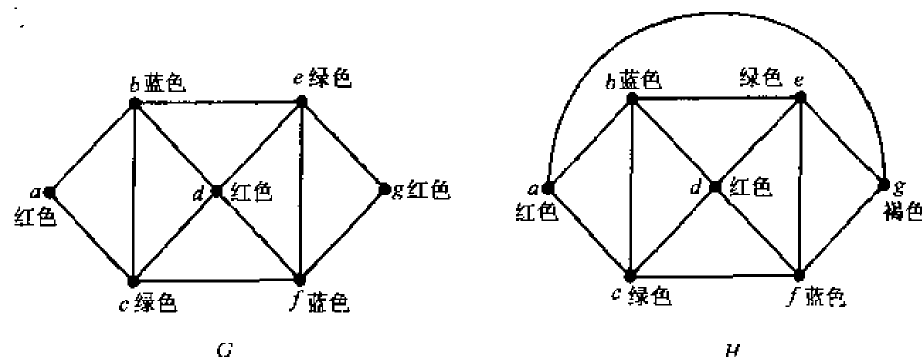


图 7-81 图 G 和 H 的着色

例 2 K_n 的色数是什么？

解 通过给每个顶点指定一种不同的颜色，用 n 种颜色可以构造 K_n 的着色。有没有使用更少颜色的着色？答案是没有。没有两个顶点可以指定相同的颜色，因为这个图的每两个顶点都是相邻的。因此，K_n 的色数 $= n$。（回忆一下，当 $n \geq 5$ 时 K_n 不是平面性图，所以这个结果与四色定理并不矛盾。）在图 7-82 中显示使用五种颜色的 K_5 的着色。■

例 3 完全偶图 $K_{m,n}$ 的色数是什么，其中 m 和 n 都是正整数？

解 需要的颜色数似乎依赖于 m 和 n。不过，仅仅需要两种颜色。用一种颜色着色 m 个顶点，而用第二种颜色着色 n 个顶点。因为边都只能连接 m 个顶点中的一个顶点与 n 个顶点中的一个顶点，所以没有相邻的顶点具有相同的颜色。在图 7-83 中显示带有两种颜色的 $K_{3,4}$ 的着色。■

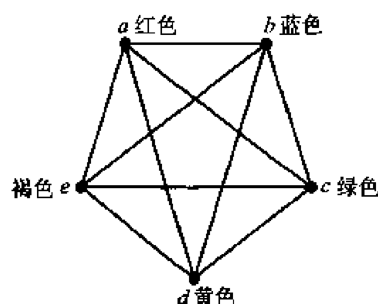


图 7-82 K_5 的着色

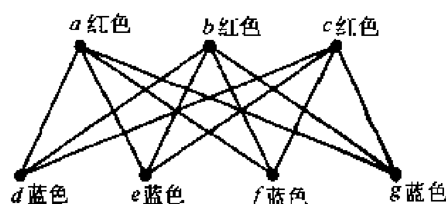


图 7-83 $K_{3,4}$ 的着色

每个连通偶简单图都具有色数 2 或 1，因为在例 3 中使用的推理适用于任何这样的图。反过来，每个色数为 2 的图都是偶图。（见本节末尾的练习 23 和练习 24。）

例 4 图 C_n 的色数是什么？（回忆一下，C_n 是带有 n 个顶点的圈图。）

解 将首先考虑一些个别情形。首先，设 $n = 6$。挑选一个顶点并且把它着色成红色。在图 7-84 中所示的 C_6 的平面画法里顺时针前进。必须给到达的下一个顶点指定第二种颜色，比如蓝色。以顺时针方向继续下去，可以用红色给第三个顶点着色，用蓝色给第四个顶点着色，用红色给第五个顶点着色。最后，可以用蓝色给第六个顶点着色，它与第一个顶点

是相邻的。因此，C_6 的色数为 2。图 7-84 显示这里构造的着色。

其次，设 $n=5$ 并且考虑 C_5。挑选一个顶点并且把它着色成红色。顺时针前进，必须给到达的下一个顶点指定第二种颜色，比如蓝色。以顺时针方向继续下去，可以用红色给第三个顶点着色，用蓝色给第四个顶点着色。用红色或蓝色都不能给第五个顶点着色，因为它与第四个顶点和第一个顶点都相邻。所以，对这个顶点就需要第三种颜色。注意，假如以逆时针方向对顶点着色，那么同样需要三种颜色。

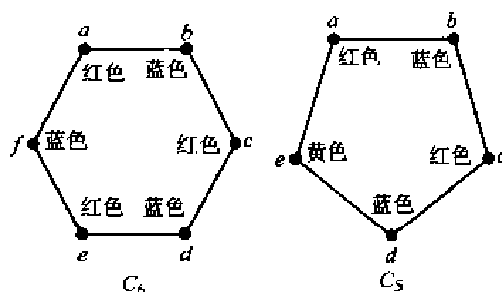


图 7-84　C_5 与 C_6 的着色

在一般情形里，当 n 是偶数时，对 C_n 着色需要两种颜色。为了构造这样的着色，简单地挑选一个顶点并且把它着色成红色。然后（利用图的平面表示）以顺时针方向绕图前进，把第二个顶点着色成蓝色，把第三个顶点着色成红色，依次类推。可以把第 n 个顶点着色成蓝色，因为与它相邻的两个顶点（即第 $n-1$ 个顶点和第一个顶点）都着色成红色。

当 n 是奇数而且 $n>1$ 时，C_n 的色数为 3。为了看出这一点，挑选一个初始顶点。为了只用两种颜色，当以顺时针方向遍历这个图时，必须使用交替的颜色。不过，所到达的第 n 个顶点与带不同颜色的两个顶点相邻，即第一个顶点和第 $n-1$ 个顶点。因此，必须使用第三种颜色。■


已知的最好的求图的色数的算法（对图的顶点数来说）具有指数的最坏情形时间复杂性。即使求图的色数的近似值的问题也是困难的。已经证明，假如存在具有多项式最坏情形时间复杂性的算法，图的色数可以达到 2 倍近似值（即构造出一个不超过图的色数的两倍的界限），那么也存在具有多项式最坏情形时间复杂性的求图的色数的算法。

7.8.2　图着色的应用

图着色对于与调度和指派有关的问题具有各种应用。在这里将给出这样的应用的例子。第一个应用处理期末考试的安排。

例 5　如何安排一所大学里的期末考试，使得没有学生在同一时间有两门考试？

解　这样的安排问题可以用图模型来解决，用顶点表示课程，若在两个顶点所表示的课程里有公共的学生，则在这两个顶点之间有边。用不同颜色来表示期末考试的每个时间段。考试的安排就对应于所关联的图的着色。

例如，假定要安排七门期末考试。假定课程编号为 1 到 7。假定下列成对的课程有公共的学生：1 和 2，1 和 3，1 和 4，1 和 7，2 和 3，2 和 4，2 和 5，2 和 7，3 和 4，3 和 6，3 和 7，4 和 5，4 和 6，5 和 6，5 和 7，以及 6 和 7。在图 7-85 里显示这组课程所关联的图。一种安排就是由这个图的一种着色来组成的。

因为这个图的色数为 4（读者应当验证这一点），所以需要 4 个时间段。在图 7-86 中显示使用了 4 种颜色的这个图的着色以及所关联的安排。■

现在考虑对电视频道分配的应用。

例 6　**频率分配**　把频道 2 到 13 分配给在北美洲的电视台，使得没有在150 英里之内

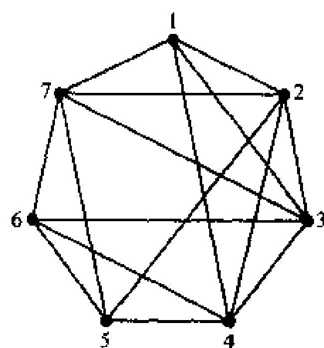
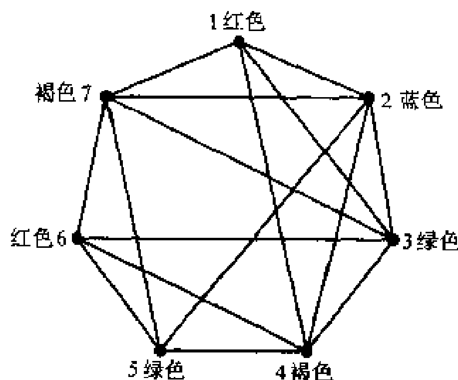


图 7-85　表示期末考试安排的图



时间段	课程
I	1,6
II	2
III	3,5
IV	4,7

图 7-86　用着色来安排期末考试

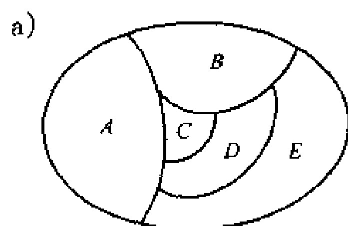
的两家电视台可以在相同频道上播出。如何用图为频道分配建模？

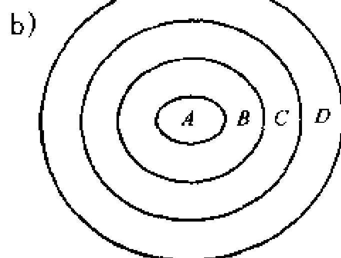
解　这样构造一个图：给每个电视台指定一个顶点。若两个电视台彼此位于 150 英里以内，则用边连接这两个顶点。频道分配就对应于这个图的着色，其中每种颜色表示一个不同的频道。■

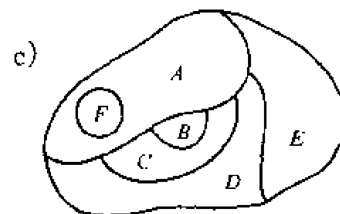
例 7　**变址寄存器**　在有效的编译器里，当把频繁地使用的变量暂时地保存在中央处理单元而不是保存在常规内存时，可以加速循环的执行。对于给定的循环来说，需要多少个变址寄存器？可以用图着色模型来讨论这个问题。为了建立这个模型，设图的每个顶点表示循环里的一个变量。若在循环执行期间两个顶点所表示的变量必须同时保存在变址寄存器里，则在这两个顶点之间有边。所以，这个图的色数就给出所需要的变址寄存器数，因为当表示变量的顶点在图中相邻时，就必须给这些变量分配不同的寄存器。■

练习

1. 构造下列每个图的对偶图。

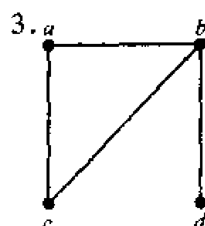
a)


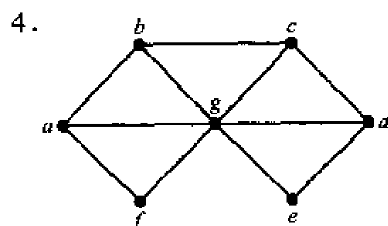
b)


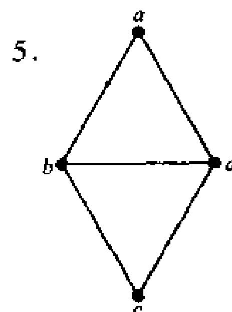
c)


2. 求练习 1 中地图着色所需要的色数，使得相邻的两个区域都没有相同的颜色。

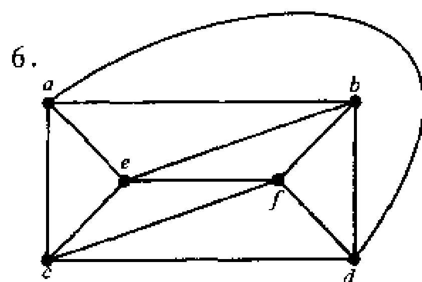
在练习 3～9 中，求所给的图的色数。

3.


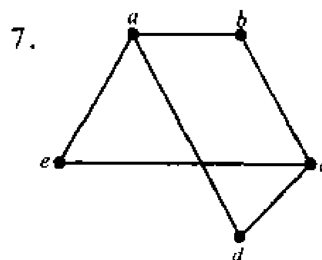
4.


5.


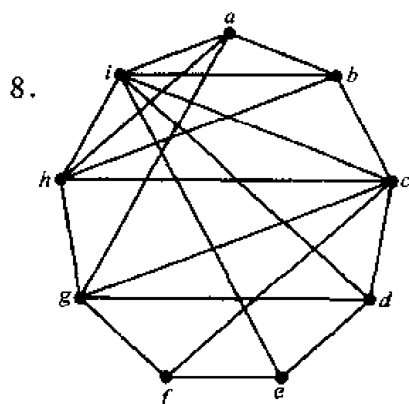
6.



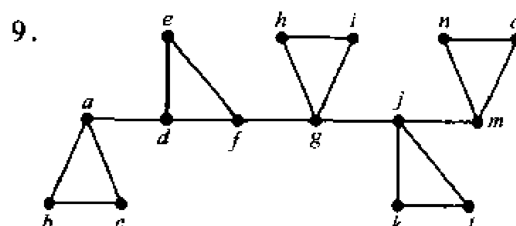
7.



8.



9.



10. 对练习3～9中的图，判定是否可能通过删除单个顶点和与它关联的所有边，来减少色数。
11. 哪些图具有色数1？
12. 为美国地图着色所需要的最少颜色数是什么？不要考虑只在一个尖角处交界的相邻州。假定密歇根是一个区域。把表示阿拉斯加和夏威夷的顶点当作孤立点。
13. W_n 的色数是什么？
14. 证明：具有包含奇数个顶点的回路的简单图不能用两种颜色来着色。
15. 假定没有学生同时选修 Math 115 与 CS 473，Math 116 与 CS 473，Math 195 与 CS 101，Math 195 与 CS 102，Math 115 与 Math 116，Math 115 与 Math 185，Math 185 与 Math 195，但是在课程的其他每种组合里都有选修的学生，请使用最少个数的不同时间段，来为 Math 115，Math 116，Math 185，Math 195，CS 101，CS 102，CS 273 和 CS 473 安排期末考试日程表。
16. 假定当两家电视台相距在150英里以内时，它们就不能使用相同的频道，那么对位于下面表中所示距离的6家电视台来说，需要多少个不同的频道？

	1	2	3	4	5	6
1	–	85	175	200	50	100
2	85	–	125	175	100	160
3	175	125	–	100	200	250
4	200	175	100	–	210	220
5	50	100	200	210		100
6	100	160	250	220	100	–

17. 数学系有六个委员会，都是每月开一次会。假定委员会是 C_1 = {阿林豪斯，布兰得，沙斯拉夫斯基}，C_2 = {布兰得，李，罗森}，C_3 {阿林豪斯，罗森，沙斯拉夫斯基}，C_4 {李，罗森，沙斯拉夫斯基}，C_5 = {阿林豪斯，布兰得}，C_6 = {布兰得，罗森，沙斯拉夫斯基}，那么必须用多少个不同的会议时间，来确保没有人被安排同时参加两个会议。
18. 动物园想建立动物自然栖息地，在里面展出它的动物。不幸的是，一些动物一有机会就会吃掉另外一些动物。如何用图模型和着色来确定所需要的不同栖息地的数目，以及在这些栖息地里的动物安置？

图的边着色是指对各边指定颜色，使得关联到相同顶点的边指定不同的颜色。图的边色数是在该图的边着色里可以使用的最少颜色数。

19. 求练习 3～9 中每个图的边色数。

*20. 求下列图的边色数。

a) K_n b) $K_{m,n}$ c) C_n d) W_n

21. 有 7 个变量出现在计算机程序的循环里。这些变量和必须保存它们的计算步骤是：t：步骤 1 到 6；u：步骤 2；v：步骤 2 到 4；w：步骤 1，3 和 5；x：步骤 1 和 6；y：步骤 3 到 6；以及 z：步骤 4 和 5。在执行期间需要多少个不同的变址寄存器来保存这些变量？
22. 关于一个以 K_n 作为子图的图的色数能说些什么？
23. 证明：色数为 2 的简单图是偶图。
24. 证明：连通偶图的色数为 2。

下面的算法可以用来为简单图着色。首先，以度递减的顺序列出顶点 v_1，v_2，…，v_n，使得 $\deg(v_1) \geqslant \deg(v_2) \geqslant \cdots \geqslant \deg(v_n)$。把颜色 1 指定给 v_1 和在表中不与 v_1 相邻的下一个顶点（若存在一个这样的顶点），并且继续指定给每一个在表中不与已经指定了颜色 1 的顶点相邻的顶点。然后把颜色 2 指定给表中还没有着色的第一个顶点。继续把颜色 2 指定给那些在表中还没有着色而且不与指定了颜色 2 的顶点相邻的顶点。若还有未着色的顶点，则指定颜色 3 给表中还没有着色的第一个顶点，并且用颜色 3 继续地对还没有着色而且不与指定了颜色 3 的顶点相邻的那些顶点着色。继续这个过程直到所有顶点都着色为止。

25. 用这个算法构造右图的着色。

*26. 用伪代码描述这个着色算法。

*27. 证明：这个算法所产生的着色可能使用了比着色一个图所需的颜色更多的颜色。

a b c d e f g h i j

图 G 的 k 重着色是对 G 的顶点指定含有 k 种不同颜色的集合，使得不把相同的颜色指定给两个相邻的顶点。用 $\chi_k(G)$ 来表示使得 G 有用 n 种颜色的 k 重着色的最小正整数 n。例如，$\chi_2(C_4) = 4$。为了看出这一点，注意如右图所示，只用四种颜色，就可以对 C_4 的

{红色，蓝色} v_1 v_2 {绿色，黄色}

{绿色，黄色} v_4 v_3 {红色，蓝色}

每个顶点指定两种颜色，使得不把相同颜色指定给两个相邻顶点。另外，少于四种颜色是不够的，因为顶点 v_1 和 v_2 每个都必须指定两种颜色，而且不能对 v_1 和 v_2 指定相同颜色。(关于 k 重着色的更多信息，参见［MiRo90］。)

28. 求下列值：

  a) $\chi_2(K_3)$　b) $\chi_2(K_4)$　c) $\chi_2(W_4)$　d) $\chi_2(C_5)$

  e) $\chi_2(K_{3,4})$　f) $\chi_3(K_5)$　*g) $\chi_3(C_5)$　h) $\chi_3(K_{4,5})$

*29. 设 G 和 H 是图 7-80 所示的图。求

  a) $\chi_2(G)$　b) $\chi_2(H)$　c) $\chi_3(G)$　d) $\chi_3(H)$

30. 若 G 是偶图而 k 是正整数，则 $\chi_k(G)$ 是什么？

31. 移动广播（或蜂窝）电话的频率是按地段分配的。每个地段分配一组该地段里的车辆所使用的频率。在产生干扰问题的地段里不能使用相同频率。解释一下如何用 k 重着色来对一个地区里的每个移动广播地段分配 k 种频率。

关键术语和结果

术语

无向边：与集合 $\{u, v\}$ 关联的边，其中 u 和 v 都是顶点

有向边：与有序对 (u, v) 关联的边，其中 u 和 v 都是顶点

多重边：连接同样一对顶点的不同的边

环：连接一个顶点与它自身的边

无向图：一组顶点以及连接这些顶点的一组无向边

简单图：没有多重边和环的无向图

多重图：可能包含多重边但不包含环的无向图

伪图：可能包含多重边和环的无向图

有向图：一组顶点以及连接这些顶点的一组有向边

有向多重图：可能包含多重有向边的带有向边的图

相邻：若在两个顶点之间有边则它们是相邻的

关联：若一个顶点是一条边的端点则那条边关联那个顶点

$\deg(v)$（无向图里顶点 v 的度）：与 v 关联的边的数目

$\deg^-(v)$（带有向边的图里顶点 v 的入度）：以 v 作为终点的边的数目

$\deg^+(v)$（带有向边的图里顶点 v 的出度）：以 v 作为起点的边的数目

带有向边的图的底图：通过忽略边的方向所获得的无向图

K_n（n 个顶点的完全图）：带 n 个顶点的无向图，其中每对顶点都用一条边连接

偶图：顶点集划分成两个子集合 V_1 和 V_2 的图，使得每条边都连接 V_1 里的顶点和 V_2 里的顶点

$K_{m,n}$（完全偶图）：顶点集划分成 m 个元素的子集和 n 个元素的子集，使得两个顶点被一条边所连接，当且仅当一个顶点属于第一个子集而另外一个顶点属于第二个子集

C_n（大小为 n 的圈图），$n \geqslant 3$：带有 n 个顶点 $v_1, v_2, \ldots, v_n$ 和边 $\{v_1, v_2\}$ $\{v_2, v_3\}$, $\ldots$, $\{v_{n-1}, v_n\}$, $\{v_n, v_1\}$ 的图

W_n（大小为 n 的轮图），$n \geq 3$：通过向 C_n 添加一个顶点以及从这个顶点到 C_n 里原来每个顶点的一条边所获得的图

Q_n（n 立方体图），$n \geq 1$：用 2^n 个长度为 n 的位串作为顶点，边连接恰好相差一位的每对位串的图

孤立点：0 度顶点

悬挂点：1 度顶点

正则图：所有顶点都有相同的度的图

图 $G=(V,E)$ 的子图：图 (W,F)，其中 W 是 V 的子图而 F 是 E 的子图

$G_1 \cup G_2$（G_1 与 G_2 的并图）：图 $(V_1 \cup V_2, E_1 \cup E_2)$，其中 $G_1=(V_1,E_1)$ 而 $G_2=(V_2,E_2)$

相邻矩阵：利用顶点的相邻关系来表示图的矩阵

关联矩阵：利用边与顶点的关联关系来表示图的矩阵

同构的简单图：对简单图 $G_1=(V_1,E_1)$ 和简单图 $G_2=(V_2,E_2)$ 来说，若存在从 V_1 到 V_2 的一一对应 f，使得对所有属于 V_1 的 v_1 和 v_2 来说，$\{f(v_1),f(v_2)\} \in E_2$ 当且仅当 $\{v_1,v_2\} \in E_1$，则 G_1 与 G_2 是同构的

不变量：同构的图都有或都没有的性质

在无向图里从 u 到 v 的通路：一条或多条边的序列 e_1，e_2，…，e_n，对 $i=0$，1，…，n 来说，e_i 关联着 $\{x_i, x_{i+1}\}$，其中 $x_0=u$ 而 $x_{n+1}=v$

在有向图里从 u 到 v 的通路：一条或多条边的序列 e_1，e_2，…，e_n，其中对 $i=0$，1，…，n 来说，e_i 关联着 (x_i, x_{i+1})，其中 $x_0=u$ 而 $x_{n+1}=v$

简单通路：不多次包含一条边的通路

回路：在相同顶点处开始与结束的通路

连通图：具有在图中每对顶点之间都有通路这个性质的无向图

连通分支：图的连通子图的集合，使得这些子图两两都没有公共点

欧拉回路：恰好包含图的每条边一次的回路

欧拉通路：恰好包含图的每条边一次的通路

哈密顿通路：简单图 $G=(V,E)$ 里的通路 x_0，x_1，…，x_n 使得 $\{x_0, x_1, \ldots, x_n\} = V$ 而且对 $0 \leq i < j \leq n$ 来说有 $x_i \neq x_j$

哈密顿回路：简单图里的回路 x_0，x_1，…，x_n，x_0，使得 x_0，x_1，…，x_n 是哈密顿通路

带权图：为各边指定数字的图

最短通路问题：确定带权图里的通路，使得这条通路里的边的权之和在所规定的顶点之间的所有通路上是最小值的问题

旅行推销员问题：求访问图的每个顶点恰好一次而总长度最短的回路的问题

平面性图：可以画在平面上而没有边交叉的图

平面性图平面表示的区域：该图的平面表示把平面分割成的区域

初等细分：删除无向图的边 $\{u, v\}$ 而且添加新顶点 w 以及边 $\{u, w\}$ 和边 $\{w, v\}$

同胚：若两个无向图是从同一个无向图通过一系列初等细分来获得的，则它们同胚

图着色：给图的顶点指定颜色，使得相邻的两个顶点没有相同的颜色

色数：在图的着色里所需要的最少颜色数

结果

在连通多重图中存在欧拉回路，当且仅当每个顶点都有偶数度。

在连通多重图中存在欧拉通路，当且仅当至多两个顶点有奇数度。

迪克斯屈拉算法：在带权图里求出两个顶点之间最短通路的过程（见 7.6.2 节）

欧拉公式：$r=e-v+2$，其中 r，e 和 v 分别是平面性图平面表示的区域数、边数和顶点数

库拉图斯基定理：图是非平面性的，当且仅当它包含同胚于 $K_{3,3}$或 K_5 的子图（其证明超出本书范围）

四色定理：每个平面图都可以用不超过四种颜色来着色（其证明远远超出本书范围！）

复习题

1. a）定义简单图、多重图、伪图、有向图、有向多重图。
   b）用例子说明如何用 a）中每种类型的图来建模。例如，解释一下如何为计算机网络或飞行航线的不同方面来建模。
2. 给出如何用图建模的至少四个例子。
3. 在无向图中顶点度数之和与该图里边数之间的关系是什么？解释为什么成立这个关系。
4. 为什么在无向图中奇数度顶点的个数必然是偶数？
5. 在有向图中顶点的入度之和与出度之和之间的关系是什么？解释为什么成立这个关系。
6. 描述下列图族。
   a）K_n，在 n 个顶点上的完全图
   b）$K_{m,n}$，在 m 和 n 个顶点上的完全偶图
   c）C_n，带 n 个顶点的圈图
   d）W_n，大小为 n 的轮图
   e）Q_n，n 立方体
7. 在练习 6 的图族中每个图有多少个顶点和多少条边？
8. a）什么是偶图？
   b）图 K_n，C_n 和 W_n 中哪些是偶图？
   c）如何确定无向图是否为偶图？
9. a）描述用来表示图的三种不同方法。
   b）画出至少带 5 个顶点和 8 条边的简单图。说明如何用你在 a）里所描述的方法来表示它。
10. a）两个简单图同构是什么意思？
    b）对于简单图的同构来说的不变量是什么意思？给出至少五个这样的不变量的例子。
    c）给出带有相同的顶点数、边数和顶点度但不是同构的两个简单图的例子。
    d）是否知道有一组不变量可以用来有效地确定两个简单图是否同构？
11. a）图是连通的是什么意思？
    b）什么是图的连通分支？
12. a）解释一下如何用相邻矩阵来表示图。
    b）如何用相邻矩阵来确定从图 G 的顶点集到图 H 的顶点集的函数是否同构？
    c）如何用图的相邻矩阵来确定在图的两个顶点之间长度为 r 的通路数？其中 r 是正整数。

13. a）定义无向图中的欧拉回路和欧拉通路。
    b）描述著名的哥尼斯堡七桥问题，并且解释一下如何利用欧拉回路来重新叙述它。
    c）如何确定无向图是否具有欧拉通路？
    d）如何确定无向图是否具有欧拉回路？
14. a）定义简单图中的哈密顿回路。
    b）给出简单图的一些性质，蕴涵着简单图没有哈密顿回路。
15. 给出至少两个可以通过求出带权图里最短通路来解决的问题的例子。
16. a）描述求在带权图两个顶点之间的最短通路的迪克斯屈拉算法。
    b）画出至少带 10 个顶点和 20 条边的带权图。用迪克斯屈拉算法求出在图中你所选择的两个顶点之间的最短通路。
17. a）图是平面性的是什么意思？
    b）给出不是平面性的简单图的例子。
18. a）平面图的欧拉公式是什么？
    b）如何用平面图的欧拉公式来证明简单图是非平面性的？
19. 叙述关于图的平面性的库拉图斯基定理，并且解释一下它如何刻画了哪些图是平面性的。
20. a）定义图的色数。
    b）当 n 是正整数时，图 K_n 的色数是什么？
    c）当 n 是大于 2 的正整数时，图 C_n 的色数是什么？
    d）当 m 和 n 都是正整数时，图 $K_{m,n}$ 的色数是什么？
21. 叙述四色定理。有没有不能用四种颜色来着色的图？
22. 解释一下在建模中可以如何使用图的着色。利用至少两个不同的例子。

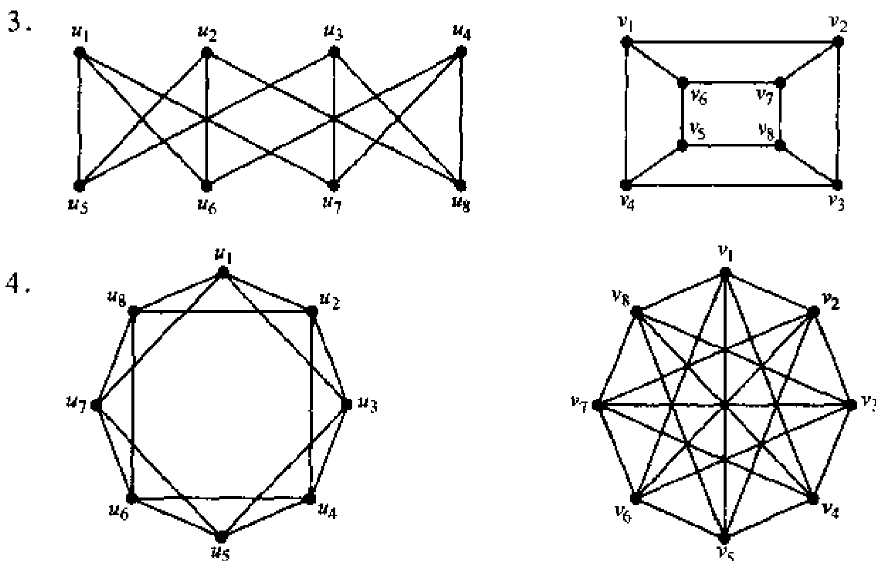
补充练习

1. 一个带 100 个顶点的 50 正则图有多少条边？
2. K_3 有多少种非同构的子图？

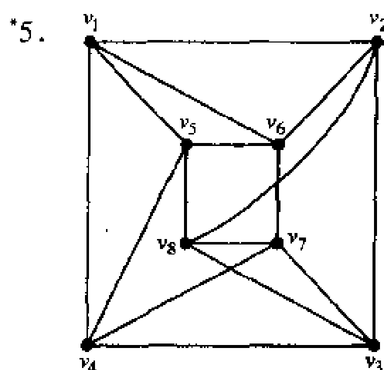
在补充练习 3～5 中，确定所给的成对的图是否同构。

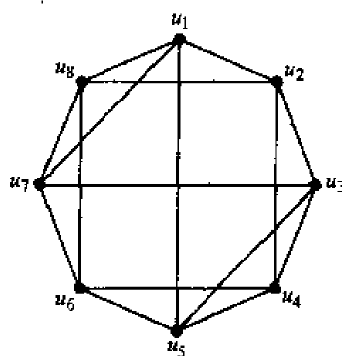
3.

4.



*5.





完全 m 部图 $K_{n_1,n_2,\dots,n_m}$ 的顶点划分成 m 个子集合，各有 n_1，n_2，…，n_m 个元素，而且顶点相邻当且仅当它们属于这个划分的不同子集合。

6. 画出下列图。

a) $K_{1,2,3}$ b) $K_{2,2,2}$ c) $K_{1,2,2,3}$

*7. 完全 m 部图 $K_{n_1,n_2,\dots,n_m}$ 有多少个顶点和多少条边？

*8. a) 证明或反驳：在至少有两个顶点的有穷简单图里，总是存在两个度数相同的顶点。

b) 对有穷多重图做与 a) 里同样的事。

设 $G=(V,E)$ 是简单图。顶点集合 V 的子集合 W 的导出子图是图 (W,F)，其中边集合 F 包含着 E 里的一条边当且仅当这条边的两个端点都属于 W。

9. 考虑第 7.4 节图 7-39 中所示的图。求下列顶点的导出子图

a) $\{a, b, c\}$

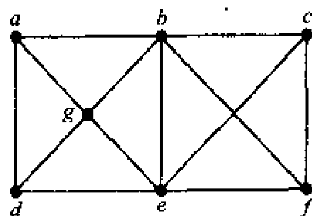
b) $\{a, e, g\}$

c) $\{b, c, f, g, h\}$

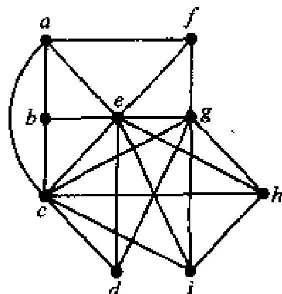
10. 设 n 是正整数。证明：K_n 的顶点集合的非空子集合的导出子图是完全图。

简单无向图里的团是一个完全子图，它不包含在任何更大的完全子图里。在补充练习 11～13 中，求所给的图的所有团。

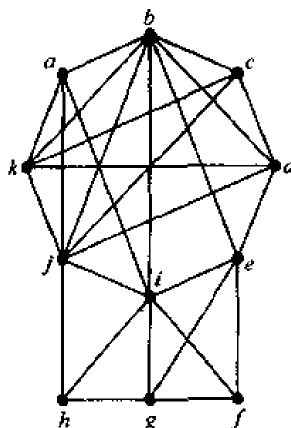
11.



12.

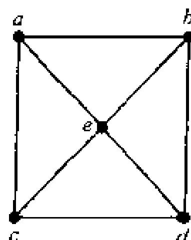


13.

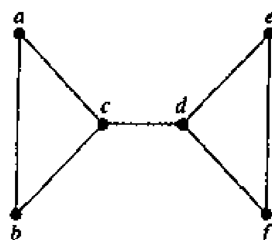


简单图里顶点的支配集是顶点的一个集合，使得其他每个顶点都与这个集合里至少一个顶点是相邻的。带最少顶点数的支配集称为最小支配集。在补充练习 14～16 中，求所给的图的最小支配集。

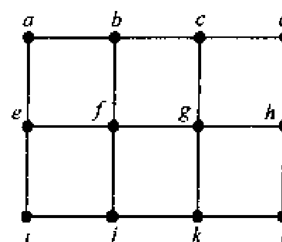
14.



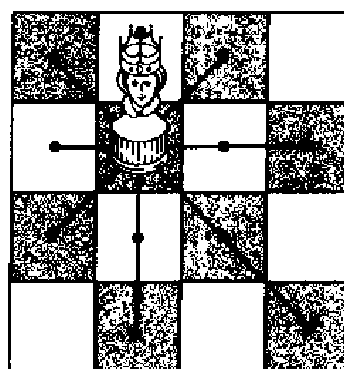
15.



16.



简单图可用来确定在棋盘上控制整个棋盘的最少皇后数。一个 $n \times n$ 的棋盘具有处在 $n \times n$ 配置下的 n^2 个格子。如图所示，在所给位置里的皇后控制着同行、同列以及包含这个格子的两条斜线上的所有格子。与此对应的简单图具有 n^2 个顶点，每个顶点表示一个格子，而且若一个顶点所表示的格子里的皇后控制着另外一个顶点所表示的格子，则这两个顶点是相邻的。


由皇后控制的方块

17. 构造表示 $n \times n$ 棋盘的简单图，用边表示皇后对格子的控制，其中

　a) $n=3$　b) $n=4$

18. 解释一下最小支配集的概念如何应用到确定控制 $n \times n$ 棋盘的最少皇后数的问题。

**19. 求控制 $n \times n$ 棋盘的最少皇后数，其中

　a) $n=3$　b) $n=4$　c) $n=5$

20. 假定 G_1 和 H_1 是同构的而且 G_2 和 H_2 是同构的。证明或反驳：$G_1 \cup G_2$ 和 $H_1 \cup H_2$ 是同构的。

21. 证明：下列性质是同构的简单图都有或都没有的不变量。

　a) 连通性

　b) 哈密顿回路的存在性

　c) 欧拉回路的存在性

　d) 有交叉数 C

　e) 有 n 个孤立顶点

　f) 是偶图

22. 如何从 G 的相邻矩阵求 $\overline{G}$ 的相邻矩阵？其中 G 是简单图。

23. 有多少种非同构的带有 4 个顶点的连通简单偶图？

*24. 有多少种非同构的简单连通图带有 5 个顶点并且

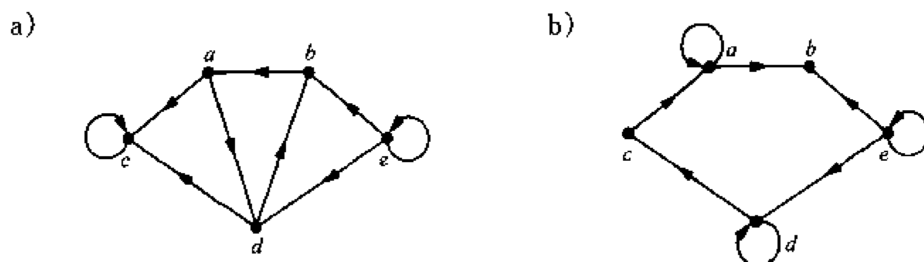
　a) 没有任何顶点的度超过 2？

　b) 色数等于 4？

　c) 是非平面性的？

若有向图与它的逆同构，则它是自逆的。

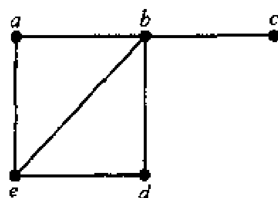
25. 确定下列图是否为自逆的。



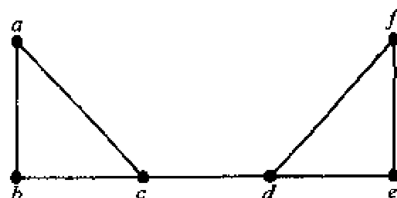
26. 证明：若有向图 G 是自逆的而且 H 是同构于 G 的有向图，则 H 也是自逆的。

无向简单图的定向就是指定它的各边的方向，使得所得到的有向图是强连通的。当无向图有定向时，这个图称为可定向的。在补充练习 27～29 中，确定给定的图是否为可定向的。

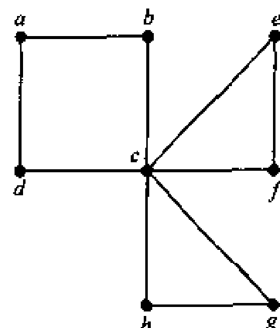
27.



28.



29.



30. 因为在城市中心区的交通流量正在增长，所以交通工程师们正在计划把目前所有双行街道都变成单行街道。解释一下如何为这个问题建模。

*31. 证明：若一个图具有割边，则它不是可定向的。

竞赛图是简单图，使得若 u 和 v 是图里不同的顶点，则(u,v)和(v,u)中恰好有一个是图中的边。

32. 有多少种不同的带 n 个顶点的竞赛图？

33. 在竞赛图里一个顶点的入度与出度之和是什么？

*34. 证明：每个竞赛图都有哈密顿通路。

35. 给定鸡群里两只鸡，其中一只占优势。这样就定义了这个鸡群的啄食次序。如何用竞赛图来为啄食次序建模？

36. 假定 G 是带 $2k$ 个奇数度顶点的连通多重图。证明：存在 k 个子图，它们的并图是 G，其中每个子图都有欧拉通路并且任何两个子图都没有公共边。[提示：向该图添加 k 条边，连接成对的奇数度顶点，并且利用这个变大了的图的欧拉回路。]

*37. 设 G 是带有 n 个顶点的简单图。G 的带宽表示成$B(G)$，它是$\max\{|i-j| \mid a_i$ 与 a_j 是相邻的$\}$在 G 的顶点$a_1, a_2, \ldots, a_n$ 的所有排列上所取的最小值。即带宽是赋给相邻顶点的下标的最大差值在顶点的所有列表上所取的最小值。求下列图的带宽。

a) K_5　b) $K_{1,3}$　c) $K_{2,3}$　d) $K_{3,3}$　e) Q_3　f) C_5

*38. 连通简单图的两个不同顶点 v_1 和 v_2 之间的距离是在 v_1 和 v_2 之间的最短通路的长度(边数)。图的半径是从顶点 v 到其他顶点的最大距离在所有顶点 v 上所取的最小值。图的直径是在两个不同顶点之间的最大距离。求下列图的半径和直径。

a) K_6　b) $K_{4,5}$　c) Q_3　d) C_6

*39. a) 证明：若简单图 G 的直径至少为 4，则它的补图 $\overline{G}$ 的直径不超过 2。

b) 证明：若简单图 G 的直径至少为 3，则它的补图 $\overline{G}$ 的直径不超过 3。

*40. 假定一个多重图有 $2m$ 个奇数度顶点。证明：任何包含该图中每条边的回路，必然至少包含 m 条边超过一次。

41. 求第 7.6 节图 7-61 中在顶点 a 与 z 之间的次短通路。

42. 设计一个算法，求简单连通带权图里两个顶点之间的最短通路。

43. 求第 7.6 节图 7-62 中在顶点 a 与 z 之间经过顶点 e 的最短通路。

44. 设计一个算法，求简单连通带权图里两个顶点之间经过第三个指定顶点的最短通路。

*45. 证明：若 G 是至少带 11 个顶点的简单图，则或者 G 不是平面性的，或者 G 的补图 $\overline{G}$ 不是平面性的。


若在图中一组顶点的集合里任何两个顶点都不相邻，则这个顶点集合称为**独立的**。图的**独立数**是图中顶点独立集里的最大顶点数。

*46. 下列图的独立数是什么？

a) K_n　b) C_n　c) Q_n　d) $K_{m,n}$

47. 证明：一个简单图里的顶点数小于或等于这个图的独立数与色数之积。

48. 证明：一个图的色数小于或等于 $v-i+1$，其中 v 是这个图里的顶点数，而 i 是这个图的独立数。

49. 假定为了生成带有 n 个顶点的随机简单图，首先选择满足 $0 \leqslant p \leqslant 1$ 的实数 p。对 $C(n,2)$ 对不同顶点中的每一对，都生成一个在 0 与 1 之间的随机数 x。若 $0 \leqslant x \leqslant p$，则用一条边连接这两个顶点；否则就不连接这两个顶点。

a) 生成带有 m 条边的图的概率是什么？其中 $0 \leqslant m \leqslant C(n,2)$。

b) 若包含的每一条边的概率为 p，则在随机生成的带有 n 个顶点的图中，边数的期望值是什么？

c) 证明：若 $p=1/2$，则以相等的概率生成每一个带 n 个顶点的简单图。

每当向简单图添加更多的边（不添加顶点）时，仍然保持的性质称为**单调递增的**，每当从简单图删除边（不删除顶点）时，仍然保持的性质称为**单调递减的**。

50. 对下列每个性质来说，确定它是否为单调递增的和确定它是否为单调递减的。

a) 图 G 是连通的。

b) 图 G 不是连通的。

c) 图 G 有欧拉回路。

d) 图 G 有哈密顿回路。

e) 图 G 是平面性的。

f) 图 G 的色数为 4。

g) 图 G 的半径为 3。

h) 图 G 的直径为 3。

51. 证明：图的性质 P 是单调递增的当且仅当图的性质 Q 是单调递减的，其中 Q 是不具有

性质 P 这个性质。

**52. 假定 P 是简单图的单调递增的性质。证明：带 n 个顶点的随机图具有性质 P 的概率，是挑选一条边属于图的概率 p 的单调非递减函数。

计算机题目

编写具有下列输入与输出的程序。

1. 给定无向图的各边所关联的顶点对，确定每个顶点的度。
2. 给定有向图的各边所关联的有序顶点对，确定每个顶点的入度和出度。
3. 给定简单图的边列表、确定这个图是否为偶图。
4. 给定图的各边所关联的顶点对，构造这个图的相邻矩阵。(产生在出现环、多重边或有向边时执行的版本。)
5. 给定图的相邻矩阵，列出这个图的各边，并且给出每条边出现的次数。
6. 给定无向图各边所关联的顶点对，以及每条边出现的次数，构造这个图的关联矩阵。
7. 给定无向图的关联矩阵，列出它的各边，并且给出每条边出现的次数。
8. 给定正整数 n，通过产生图的相邻矩阵来生成无向图，使得以相等的概率来生成所有的简单图。
9. 给定正整数 n，通过产生图的相邻矩阵来生成有向图，使得以相等的概率来生成所有的有向图。
10. 给定两个都带不超过六个顶点的简单图的边列表，确定这两个图是否同构。
11. 给定图的相邻矩阵和正整数 n，求顶点两两之间长度为 n 的通路数。(产生对有向图和无向图来说都能工作的程序。)

*12. 给定简单图的边列表，确定它是否连通，若它不连通，则求连通分支数。

13. 给定多重图的各边所关联的顶点对，确定它是否有欧拉回路，若没有欧拉回路，则确定它是否有欧拉通路。若存在欧拉通路或欧拉回路，则构造这样的通路或回路。

*14 . 给定有向多重图的各边所关联的有序顶点对，若存在欧拉通路或欧拉回路，则构造这样的通路或回路。

**15. 给定简单图的边列表，产生一条哈密顿回路，或者确定该图没有这样的回路。

**16. 给定简单图的边列表，产生一条哈密顿通路，或者确定该图没有这样的通路。

17. 给定带权连通简单图的边及其权的列表，以及该图中的两个顶点，用迪克斯屈拉算法求这两点间最短通路的长度。另外，求出这条通路。
18. 给定无向图的边的列表，用第 7.8 节练习中所给的算法求这个图的着色。
19. 给定学生及其注册课程的表，构造期末考试日程表。
20. 给定各对电视台之间的距离，为这些台分配频率。

计算和研究

利用计算程序或你所编写的程序来做下面的练习。

1. 显示带 4 个顶点的所有简单图。
2. 显示全套的带 6 个顶点的所有非同构的简单图。

3. 显示全套的带 4 个顶点的所有有向图。
4. 随机地生成 10 个不同的简单图，每个带 20 个顶点，使得每个这样的图都是以相等的概率来生成的。
5. 构造一种格雷码，其中码字都是长度为 6 的位串。
6. 构造马在不同大小的棋盘上的巡回路线。
7. 确定你在本组练习的练习 4 中生成的每个图是否为平面性图。若可以做到，则确定每个非平面性图的厚度。
8. 确定你在本组练习的练习 4 中生成的每个图是否连通。若一个图不连通，则确定这个图的连通分支数。
9. 随机地生成带 10 个顶点的简单图。当你生成了一个带欧拉回路的图时停止。显示这个图里的一个欧拉回路。
10. 随机地生成带 10 个顶点的简单图。当你生成了一个带哈密顿回路的图时停止。显示这个图里的一个哈密顿回路。
11. 求你在本组练习的练习 4 中所生成的每个图的色数。

**12. 求旅行推销员访问美国 50 个州的每个首府所能采取的最短路线，乘飞机以直线在城市之间旅行。

*13. 对每个不超过 10 的正整数 n 来说，估计随机生成的带 n 个顶点的简单图连通的概率，方法是生成一组随机简单图并且确定每个图是否连通。

**14. 研究这样一个问题：确定 $K(7,7)$ 的交叉数是否为 77、79 或 81。已知它等于这三个数当中的一个。

写作题目

利用本书以外的资料，就下列问题写作短文。

1. 描述一下在 1900 年以前图论的起源和发展。
2. 讨论一下图论对生态系统研究的应用。
3. 讨论一下图论对社会学和心理学的应用。
4. 描述一下给定一个图的顶点和边，在纸面或屏幕上画出这个图的算法。为了使图具有显现其性质的最佳形态，画图时需要考虑什么因素？
5. 一个输入、显示和操纵各种图的软件工具应当具有什么能力？现有的工具都具有这些能力中的哪些？
6. 描述一下确定两个图是否同构的一些可用算法和这些算法的计算复杂性。目前已知最有效的算法是什么？
7. 定义德布鲁因（de Bruijn）序列并且讨论一下它们如何出现在应用里。解释一下如何用欧拉回路来构造德布鲁因序列。
8. 描述一下中国邮递员问题并且解释如何解决这个问题。
9. 描述一下蕴涵着图具有哈密顿回路的一些不同条件。
10. 描述一下用来解决旅行推销员问题的几个不同策略和算法。
11. 描述一下确定一个图是否为平面性图的几个不同算法。每个算法的计算复杂性是什么？
12. 在建模中，大规模集成电路图有时嵌入在书中，顶点在书脊上而边在书页上。定义一下

图的书数并且对 $n=3$、4、5 和 6 求包括 K_n 的各种图的书数。

13. 描述一下四色定理的历史。
14. 描述一下在四色定理的证明中计算机所扮演的角色。如何可以肯定一个依赖计算机的证明是正确的?
15. 就是否产生最少颜色的着色而言，以及就复杂性而言，描述并比较一下给图着色的几个不同算法。
16. 解释一下在各种不同模型里如何使用图的多重着色。
17. 解释一下在带特定性质的图的非构造性存在性证明中如何使用随机图理论。

第 8 章　树

不包含简单回路的连通图称为树。早在 1857 年就有人使用过树，当时英国数学家亚瑟·凯莱用它们去计数某些类型的化合物。本章里的例子将说明，从那时起，树已经被用来解决各种各样学科分支里的问题。

树在计算机科学里特别有用。例如，树用来构造在表中求出项的位置的有效算法。它们被用来构造以最便宜的电话线连接分布式计算机的网络。可用树构造存储和传输数据的有效编码。树可以用来为通过一系列决策完成的过程建立模型。这个用途使得树在排序算法的研究中很有价值。

8.1　介绍树

在图 8-1 中显示伯努利家族的族谱图，这是瑞士数学家的著名家族。这样的图也称为家族树。家族树是一个图，其中顶点表示家族成员，边表示亲子关系。表示族谱图的无向图是一种特殊类型的图的例子，这种图称为树。

定义 1　树是没有简单回路的连通无向图。

因为树没有简单回路，所以树不含多重边或环。因此任何树都必然是简单图。

例 1　在图 8-2 所示的图中，哪些图是树？

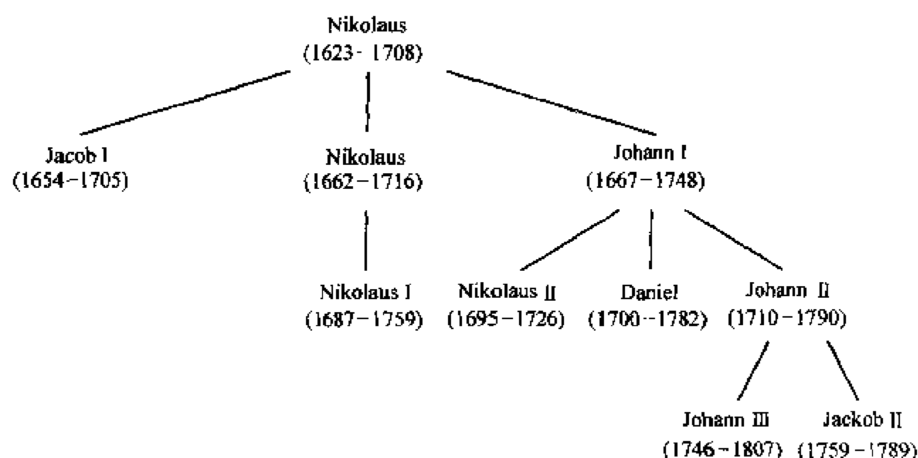


图 8-1　数学家的伯努利家族

解　G_1 和 G_2 是树，因为都是没有简单回路的连通图。G_3 不是树，因为 e, b, a, d, e 是这个图中的简单回路。最后，G_4 不是树，因为它不连通。■

任何一个不包含简单回路的连通图都是树。不含简单回路但不一定连通的图是什么？这些图称为**森林**，而且具有这样的性质：它们的每个连通分支都是树。图 8-3 显示一个森林。

通常把树定义成具有在每对顶点之间存在唯一简单通路的性质的无向图。下面的定理说明这个替代定义等价于原来的定义。

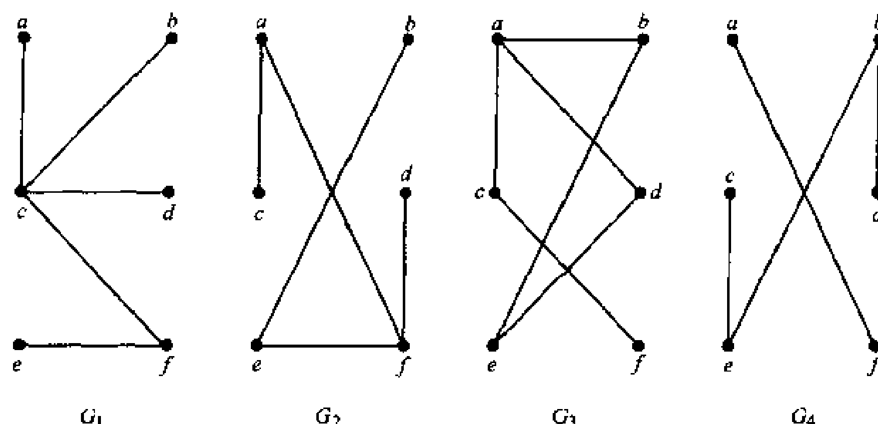


图 8-2 G_1 和 G_2 是树，G_3 和 G_4 不是树

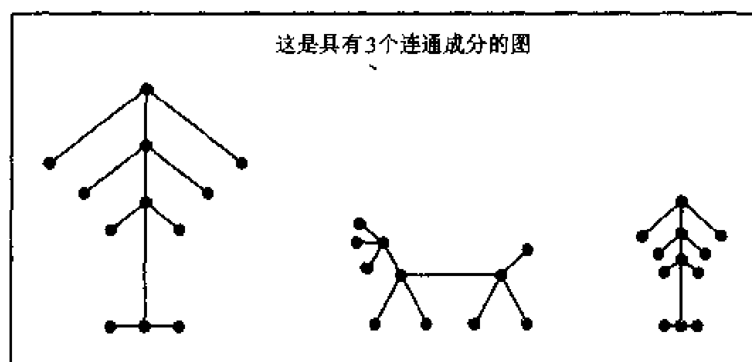


图 8-3 一个森林的例子

定理 1 一个无向图是树，当且仅当在它的每对顶点之间存在唯一简单通路。

证 首先假定 T 是树。则 T 是没有简单回路的连通图。设 x 和 y 是 T 的两个顶点。因为 T 是连通的，根据 7.4 节定理 1，在 x 和 y 之间存在一条简单通路。另外，这条通路必然是唯一的，因为假如存在第二条这样的通路，则组合从 x 到 y 的第一条这样的通路以及经过倒转从 x 到 y 的第二条通路的顺序所得到的从 y 到 x 的通路，这样将形成回路。利用 7.4 节练习 35，这蕴涵着在 T 中存在简单回路。因此，在树的任何两个顶点之间存在唯一简单通路。

现在假定在图 T 的任何两个顶点之间存在唯一简单通路。则 T 是连通的，因为在它的任何两个顶点之间存在通路。另外，T 没有简单回路。为了看出这句话是真的，假定 T 有包含顶点 x 和 y 的简单回路。则在 x 和 y 之间就有两条简单通路，因为这条简单回路包含一条从 x 到 y 的简单通路和一条从 y 到 x 的简单通路。因此，在任何两个顶点之间存在唯一简单通路的图是树。 ■

在树的许多应用里，指定树的一个特殊顶点作为根。一旦规定了根，就可以给每条边指定方向如下。因为从根到图的每个顶点存在唯一通路（根据定理 1），所以指定每条边是离开根的方向。因此，树与它的根一起产生一个有向图，称为根树。通过选择任何一个顶点来作为根，就可以把非根树变成根树。注意对根的不同选择导致产生不同的根树。例如，图 8-4 显示通过在树 T 里分别指定 a 和 c 作为根所形成的根树。通常在画根树时把根画在图的顶端。指示根树中边的方向的箭头可以省略，因为对根的选择确定了边的方向。

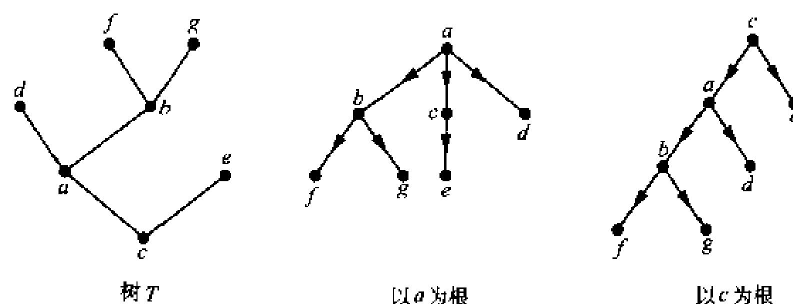


图 8-4 树和指定两个根形成的根树

树的术语起源于植物学和族谱学。假定 T 是根树。若 v 是 T 里非根的顶点，则 v 的父亲是使得从 u 到 v 存在有向边的唯一的顶点 u（读者应当证明这样的顶点 u 是唯一的）。当 u 是 v 的父亲时，v 称为 u 的儿子。具有相同父亲的顶点称为*兄弟*。非根顶点的祖先是从根到该顶点的通路上的顶点，不包括该顶点自身但包括根（即该顶点的父亲、该顶点的父亲的父亲、等等，一直到根为止）。顶点 v 的*后代*是以 v 作为祖先的顶点。树的顶点若没有儿子则称为*树叶*。有儿子的顶点称为*内点*。根是内点，除非它是图中唯一的顶点，在这种情况下它是树叶。

若 a 是树里的顶点，则以 a 为根的*子树*是由 a 和 a 的后代以及这些顶点所关联的边所组成的该树的子图。

例 2 在图 8-5 所示的根树里（有根 a），求 c 的父亲，g 的儿子，h 的兄弟，e 的所有祖先，b 的所有后代，所有内点，以及所有树叶。什么是根在 g 处的子树?

解 c 的父亲是 b。g 的儿子是 h，i 和 j。h 的兄弟是 i 和 j。e 的祖先是 c，b 和 a。b 的后代是 c，d 和 e。内点是 a，b，c，g，h 和 j。树叶是 d，e，f，i，k，l 和 m。根在 g 处的子树如图 8-6 所示。■

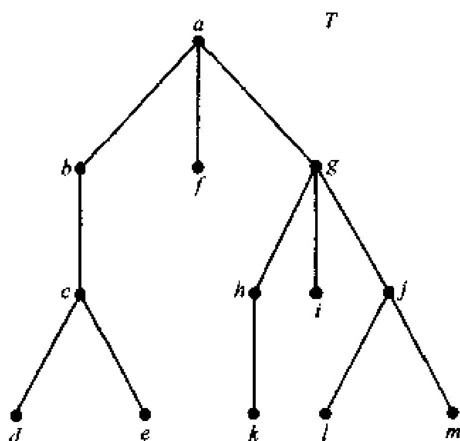


图 8-5 根树 T

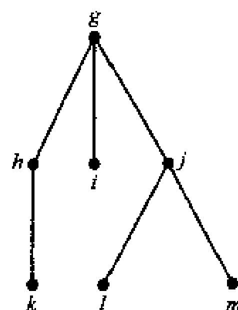


图 8-6 根在 g 处的子树

在许多不同的应用里都使用具有下面性质的根树：它们的所有内点都有同样个数的儿子。在本章后面将用这样的树去研究涉及到搜索、排序和编码的问题。

定义 2 若根树的每个内点都有不超过 m 个儿子，则称它为 m 元树。若该树的每个内

点都恰好有 m 个儿子，则称它为满 m 元树。把 $m=2$ 的满 m 元树称为二叉树。

例 3 在图 8-7 里的根树，是否对某个正整数 m 来说是满 m 元树？

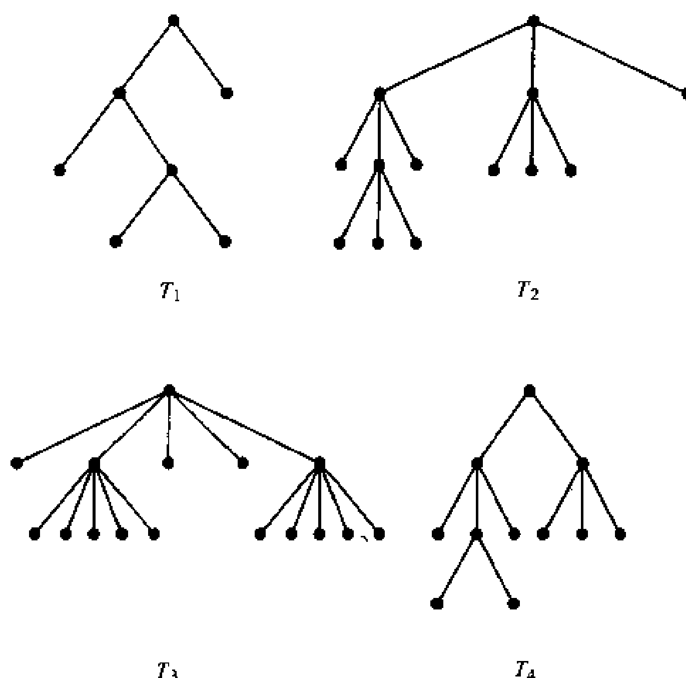


图 8-7 四个根树

解 T_1 是满二叉树，因为它的每个内点都有两个儿子。T_2 是满三元树，因为它的每个内点都有三个儿子。在 T_3 里每个内点都有五个儿子，所以它是满五元树。对任何 m 来说 T_4 都不是满 m 元树，因为它的某些内点有两个儿子而其他内点有三个儿子。 ■

*有序根树*是把每个内点的儿子们都排序的根树。画有序根树时，以从左向右的顺序来显示每个内点的儿子们。注意在常规方式里，根树的表示确定它的边的一种顺序。将在画图时使用边的这种顺序，而不明确地指出认为根树是有序的。在有序二叉树中，若一个内点有两个儿子，则第一个儿子称为*左儿子*而第二个儿子称为*右儿子*。根处在一个顶点的左儿子处的树称为该顶点的*左子树*，而根处在一个顶点的右儿子处的树称为该顶点的*右子树*。读者应当注意，对某些应用来说，二叉树的每个非根顶点都指定为其父亲的右儿子或左儿子。即使当某些顶点仅有一个儿子时也这样做。每当有必要时就作出这样的指定，但是没有必要时就不这样做。

例 4 在图 8-8 a) 所示二叉树 T 里，什么是 d 的左儿子和右儿子（其中顺序是画法所蕴涵的）？什么是 c 的左子树和右子树。

解 d 的左儿子是 f 而右儿子是 g。在图 8-8 b) 和图 8-8 c) 中分别显示 c 的左子树和右子树。 ■

与图的情形恰好一样，不存在用来描述树、根树、有序根树和二叉树等的标准的术语。出现这种非标准的术语，是因为在计算机科学里大量地使用树，而计算机科学还是相对年轻的领域。每当出现关于树的术语时，读者就应当仔细地核对这些术语所表示的意思。

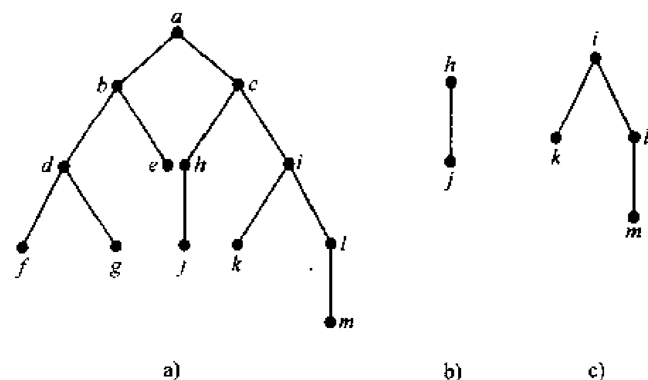


图 8-8 二叉树 T 和顶点 c 的左子树和右子树

8.1.1 树作为模型

树在如此广泛的领域里用来作为模型，比如计算机科学、化学、地质学、植物学和心理学等。我们将描述基于树的各式各样的模型。

例 5 饱和碳氢化合物与树 图可以用来表示分子，其中用顶点表示原子，用边表示原子之间的化学键。英国数学家亚瑟·凯莱[⊖]在 1857 年发现了树，当时他正在试图列举形如 C_nH_{2n+2} 的化合物的同分异构体，它们都称为饱和碳氢化合物。

在饱和碳氢化合物的图模型里，用 4 度顶点表示每个碳原子，用 1 度顶点表示每个氢原子。在形如 C_nH_{2n+2} 的化合物的表示图里有 $3n+2$ 个顶点。在这个图中边数是顶点度数之和的一半。因此，在这个图中有 $(4n+2n+2)/2=3n+1$ 条边。因为这个图是连通的，而且边数比顶点数少 1，所以它必然是树（见本节末的练习 9）。

带有 n 个 4 度顶点和 $2n+2$ 个 1 度顶点的非同构的树，就表示 C_nH_{2n+2} 的不同的同分异构体。例如，当 $n=4$ 时，存在恰好两个 C_4H_{10} 的不同的同分异构体。它们的结构显示在图 8-9 中。这两种同分异构体称为丁烷和异丁烷。■

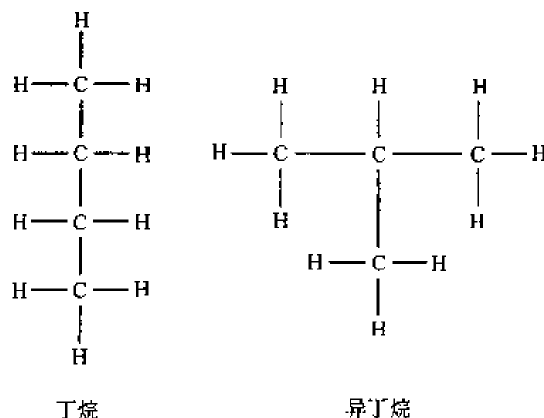


图 8-9 丁烷的两种同分异构体

⊖ 亚瑟·凯莱（Arthur Cayley，1821—1895） 凯莱是一个商人的儿子，他在很小年纪就以数字计算的惊人技巧显示出他的数学天才。凯莱在 17 岁时进入剑桥的三一学院。在学院期间他培养出对阅读小说的爱好。凯莱在剑桥表现优秀，被选举为任期三年的三一研究员和助教。在这期间凯莱开始了对 n 维几何学的研究，对几何学和分析学作出了多种贡献。他还培养出对登山的兴趣，在瑞士度假时就以登山为乐。因为没有合适的数学家职位给他，所以凯莱离开剑桥大学，进入法律行业并且在 1849 年获得律师资格。尽管凯莱限制他的法律工作以便继续他的数学研究，他仍然赢得了作为法律专家的声誉。在他的法律生涯中，他努力写出了超过 300 篇的数学论文。剑桥大学在 1863 年设立一个新的数学职位并且把它给了凯莱。他接受了这个工作，虽然它的报酬低于他作为律师所获得的报酬。

例 6 *表示组织机构* 大的组织机构的结构可以用根树来建模。在这个树里每个顶点表示机构里的一个职务。从一个顶点到另外一个顶点的边的始点所表示的人是终点所表示的人的（直接）上司。图 8-10 所示的图就表示这样的树。在这个树所表示的组织机构里，硬件开发主任直接为研发经理工作。■

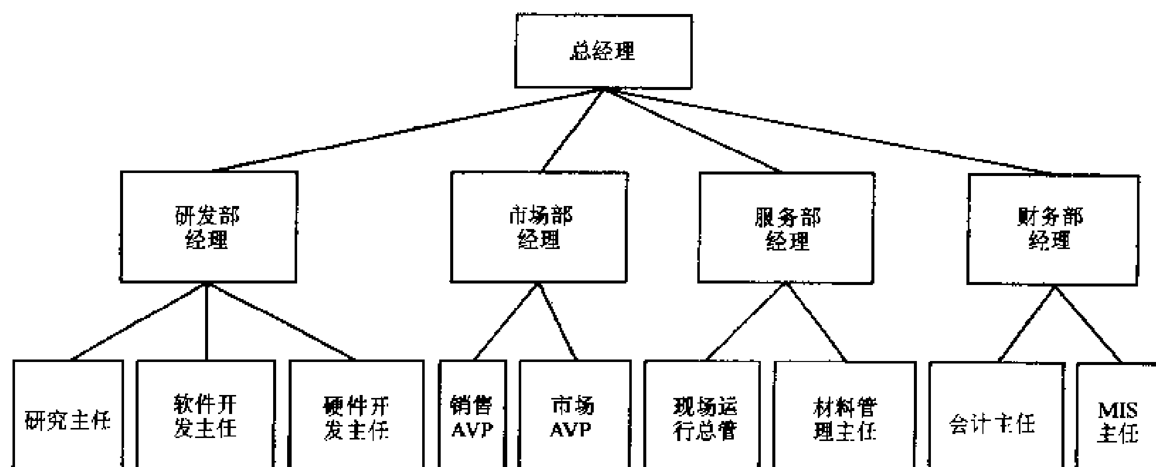


图 8-10 一家计算机公司的组织机构图

例 7 *计算机文件系统* 计算机存储器中的文件可以组织成目录。目录可以包含文件和子目录。根目录包含整个文件系统。因此，文件系统可以表示成根树，其中根表示根目录，内点表示子目录，树叶表示文件或空目录。在图 8-11 中显示一个这样的文件系统。在该系统中，文件 khr 属于目录 rje。■

例 8 *树形连接并行处理系统* 在 7.2 节例 13 里描述过多种并行处理的互联网络。树形连接网络是把处理器互相连接的另外一种重要方式。这样的网络的表示图是满二叉树。这样的网络把 $n=2^k-1$ 个处理器互连起来，其中 k 是正整数。一个非根也非树叶的顶点 v 所表示的处理器具有三个双向连接，一个连接通向 v 的父亲所表示的处理器，两个连接通向 v 的两个儿子所表示的处理器。根所表示的处理器具有两个双向连接，分别通向它的两个儿子所表示的处理器。树叶所表示的处理器具有一个双向连接，通向它的父亲。在图 8-12 里显示带 7 个处理器的树形连接网络。

将要说明并行计算是如何使用树形连接网络的。具体地说，将说明图 8-12 里的处理器是如何用三步来相加 8 个数的。在第一步用 P_4 相加 x_1 和 x_2。用 P_5 相加 x_3 和 x_4。用 P_6 相加 x_5 和 x_6。用 P_7 相加 x_7 和 x_8。在第二步用 P_2 相加 x_1+x_2 和 x_3+x_4。用 P_3 相加 x_5+x_6 和 x_7+x_8。最后，在第三步用 P_1 相加 $x_1+x_2+x_3+x_4$ 和 $x_5+x_6+x_7+x_8$。用来相加 8 个数的这三步，优于串行地相加 8 个数所需要的七步，串行的步骤是依次把一个数与表中前面各数之和相加。■

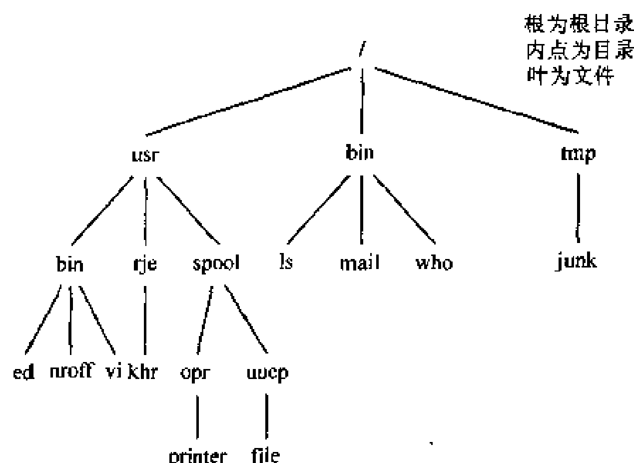


图 8-11 一个计算机文件系统

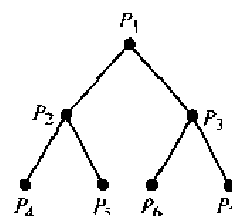


图 8-12 带 7 个处理器的树形连接网络

8.1.2 树的性质

将常常需要那些建立树里各种各样的边和顶点的数目之间的关联的结果。

定理 2 带有 n 个顶点的树含有 $n-1$ 条边。

证 选择顶点 r 作为树的根。让每条边的终点都与这条边关联，在边与除 r 外的顶点之间建立起一一对应。因为除 r 外还有 $n-1$ 个顶点，所以树中有 $n-1$ 条边。 □

下面的定理说明，带有指定的内点数的满 m 元树的顶点数是确定的。与定理 2 一样，将用 n 来表示树中的顶点数。

定理 3 带有 i 个内点的满 m 元树含有 $n=mi+1$ 个顶点。

证 除根之外的每个顶点都是内点的儿子。因为每个内点有 m 个儿子，所以在树中除根之外还有 mi 个顶点。因此，该树含有 $n=mi+1$ 个顶点。 □

假定 T 是满 m 元树。设 i 是该树的内点数而 l 是树叶数。一旦 n，i 和 l 之中的一个是已知的，另外的两个量就随之确定了。在下面的定理中，给出如何从已知的一个量求其他两个量的方法。

定理 4 一个满 m 元树带有

(i) n 个顶点有 $i=(n-1)/m$ 个内点和 $l=[(m-1)n+1]/m$ 个树叶。

(ii) i 个内点有 $n=mi+1$ 个顶点和 $l=(m-1)i+1$ 个树叶。

(iii) l 个树叶有 $n=(ml-1)(m-1)$个顶点和 $i=(l-1)/(m-1)$个内点。

证 设 n 表示顶点数，i 表示内点数，l 表示树叶数。利用定理 3 中的等式，即 $n=mi+1$，以及等式 $n=l+I$（这个等式为真，是因为每一个顶点要么是树叶、要么是内点），就可以证明本定理的所有三个部分。将在这里证明部分（i）。部分（ii）和（iii）的证明留给读者作为习题。

在 $n=mi+1$ 里求解 I 得出 $i=(n-1)/m$。然后把 i 的这个表达式代入等式 $n=l+i$，就证明 $l=n-i=n-(n-1)/m=[(m-1)n+1]/m$。 □

下面的例子说明如何使用定理 4。

例 9　假定某人寄出一封连环信。要求收到信的每个人再把它寄给另外 4 个人。有一些人这样做了，但是其他人则没有寄出信。若没有人收到超过一封的信，而且若在有 100 个人读过信但是不寄出它之后，连环信就终止了，则包括第一个人在内，有多少人看过信？有多少人寄出过信？

解　可以用 4 元树表示连环信。内点对应于寄出信的人，而树叶对应于不寄出信的人。因为有 100 个人不寄出信，所以在这个根树里，树叶数是 $l = 100$。因此，定理 4 的部分 (iii) 说明，已经看过信的人数是 $n = (4 \cdot 100 - 1)/(4 - 1) = 133$。另外，内点数是 $133 - 100 = 33$，所以 33 个人寄出过信。■

经常需要使用这样的根树，它们是“平衡的”，所以在每个顶点处的子树都包含大约相同长度的通路。下面的一些定义将解释清楚这个概念。在根树里顶点 v 的*层数*是从根到这个顶点的唯一通路的长度。根的层数定义为 0。根树的*高度*就是顶点层数的最大值。换句话说，根树的层数是从根到任意顶点的最长通路的长度。

例 10　求图 8-13 所示的根树里每个顶点的层数。这棵树的高度是什么？

解　根 a 在 0 层上。顶点 b，j 和 k 都在 1 层上。顶点 c，e，f 和 l 都在 2 层上。顶点 d，g，i，m 和 n 都在 3 层上。最后，顶点 h 在 4 层上。因为任意顶点的最大层数是 4，所以这棵树的高度为 4。■

若一棵高度为 h 的根 m 元树的所有顶点都在 h 层或 $h-1$ 层，则这棵树是平衡的。

例 11　在图 8-14 所示的一些根树里，哪些根树是平衡的？

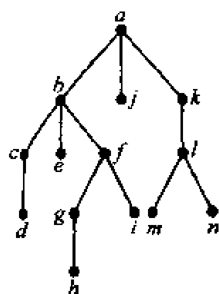


图 8-13　一棵根树

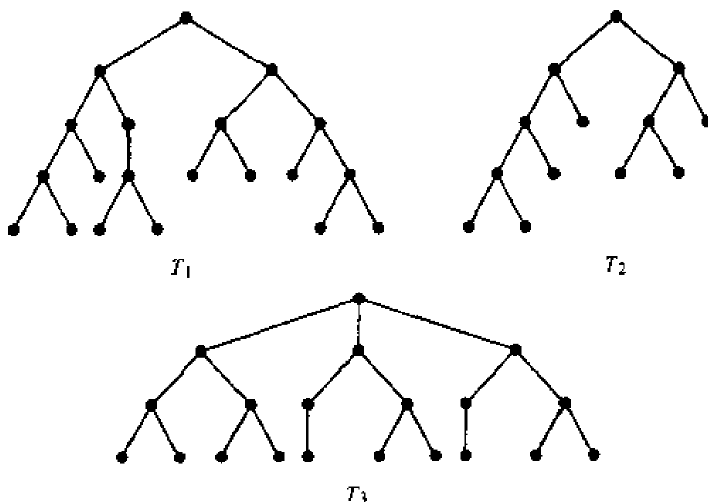


图 8-14　一些根树

解　T_1 是平衡的，因为它所有的树叶都在 3 层和 4 层上。不过，T_2 不是平衡的，因为它有树叶在 2 层、3 层和 4 层上。最后，T_3 是平衡的，因为它所有的树叶都在 3 层上。■

下面的结果建立 m 元树的高度与叶数之间的联系。

定理 5　在高度为 h 的 m 元树里至多有 m^h 个树叶。

证　本证明对高度使用数学归纳法。首先，考虑高度为 1 的 m 元树。这些树都是由带

有不超过 m 个儿子的一个根所组成的，每个儿子都是树叶。因此在高度为 1 的 m 元树里有不超过 $m^1 = m$ 个树叶。这是归纳论证的基础步骤。

现在假定对高度小于 h 的所有 m 元树来说这个结果都为真；这是归纳假设。设 T 是高度为 h 的 m 元树。T 的树叶都是通过删除从根到每个在 1 层的顶点的边所获得的 T 的各子树的树叶，如图 8-15 所示。

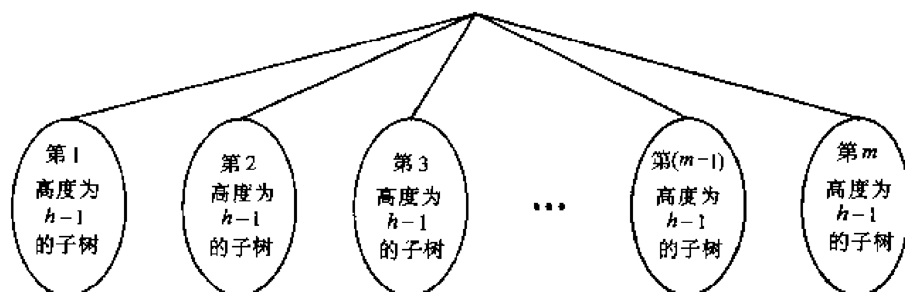


图 8-15　证明的归纳步骤

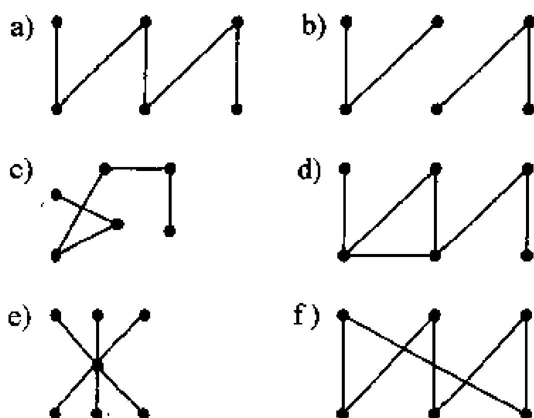
这些子树每个高度都小于或等于 $h-1$。所以根据归纳假设，这些根树每个都有至多 m^{h-1}条边。因为最多有 m 个这样的子树，每个子树最多有 m^{h-1}个树叶，所以在这个根树里最多有 $m \cdot m^{h-1} = m^h$ 个树叶。这样就完成了归纳论证。□

推论 1　若一个高度为 h 的 m 元树带有 l 个树叶，则 $h \geqslant \lceil \log_m l \rceil$。若这个 m 元树是满的和平衡的，则 $h = \lceil \log_m l \rceil$。（在这里使用上取整函数。回忆一下，$\lceil x \rceil$是大于或等于 x 的最小整数。）

证　从定理 5 知道 $1 \leqslant m^h$。取以 m 为底的对数就证明 $h \geqslant \log_m l$。因为 h 是整数，所以有 $h \geqslant \lceil \log_m l \rceil$。现在假定这个树是平衡的。于是每个树叶都在 h 层或 $h-1$ 层上，而且因为树的高度为 h，所以在 h 层至少有一个树叶。所以必然有超过 m^{h-1}个树叶（见本节末尾的练习 24）。因为 $l \leqslant m^h$，所以有 $m^{h-1} < l \leqslant m^h$。在这个不等式里取以 m 为底的对数就得出 $h-l < \log_m l \leqslant h$。因此 $h = \lceil \log_m l \rceil$。□

练习

1．下面哪些图是树？



2. 回答下列关于如图所示的根树的问题。

a) 哪个顶点是根？

b) 哪些顶点是内点？

c) 哪些顶点是树叶？

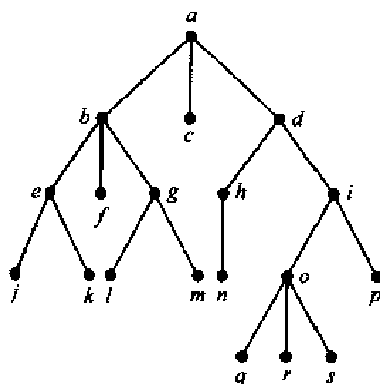
d) 哪些顶点是 i 的儿子？

e) 哪些顶点是 h 的父亲？

f) 哪些顶点是 o 的兄弟？

g) 哪些顶点是 m 的祖先？

h) 哪些顶点是 b 的后代？



3. 习题 2 里的根树是否对某个正整数 m 来说是满 m 元树？

4. 习题 2 里的树的每个顶点的层数是什么？

5. 画出练习 2 里的树以下列顶点为根的子树。

a) a b) c c) e

*6. 有多少种非同构的带有 n 个顶点的非根树？若

a) $n=3$ b) $n=4$ c) $n=5$

*7. 对根树回答与练习 6 所给的问题相同的问题（使用有向图的同构）。

*8. 证明：简单图是树，当且仅当它是连通的，但是删除它的任何一条边就产生不连通的图。

*9. 设 G 是带有 n 个顶点的简单图。证明：G 是树，当且仅当 G 是连通的并且有 $n-1$ 条边。

10. 哪些完全偶图 $K_{m,n}$ 是树？其中 m 和 n 都是正整数。

11. 带有 10 000 个顶点的树有多少条边？

12. 带有 100 个内点的满 5 元树有多少个顶点？

13. 带有 1 000 个内点的满二叉树有多少条边？

14. 带有 100 个顶点的满 3 元树有多少个树叶？

15. 假定 1 000 个人参加象棋巡回赛。若一个选手输掉一盘就遭到淘汰，而且比赛进行到只有一位参加者还没有输过为止，则利用这个巡回赛的根树模型，来确定为了决出冠军必须下多少盘棋？（假定没有平局。）

16. 一次连环信开始时有一个人寄出一封信给其他 5 个人。收到这信的每个人或者寄出这信

给从来没有收到过它的其他 5 个人，或者不把它寄给任何人。假定在这个连环终止以前有 10 000 人寄出过这信，并且没有人收到过超过一封信。有多少人收到过信？又有多少人收到过信但是没有寄出过它？

17. 一次连环信开始时一个人寄出一封信给其他 10 个人。要求每个人寄出这信给其他 10 个人，而且每封信都包含该连环里前面 6 个人的列表。除非表中不足 6 个名字，否则每个人都寄 1 美元给表中第一个人，从表中删除这个人的名字，把其他 5 个人的名字向上移动一位，并且把他或她自己的名字插入到表的末尾。若没有人中断这个连环，并且没有人收到超过一封信，则这个连环里的一个人最终将收到多少钱？

*18. 要么画出带有 76 个树叶而且高度为 3 的满 m 元树，其中 m 是正整数，要么证明不存在这样的树。

*19. 要么画出带有 84 个树叶而且高度为 3 的满 m 元树，其中 m 是正整数，要么证明不存在这样的树。

*20. 一个满 m 元树 T 有 81 个树叶并且高度为 4。

a) 给出 m 的上界和下界。

b) 若 T 也是平衡的，则 m 是什么？

一个完全 m 元树是其中每个树叶都在同一层上的满 m 元树。

21. 构造高度为 4 的完全二叉树和高度为 3 的完全 3 元树。

22. 高度为 h 的完全 m 元树具有多少个顶点和多少个树叶？

23. 证明：

a) 定理 4 的部分（ii）。

b) 定理 4 的部分（iii）。

☞ 24. 证明：高度为 h 的满 m 元树具有超过 m^{h-1}个树叶。

25. 在包含总共 n 个顶点的 t 个树的森林里有多少条边？

26. 解释一下如何用树来表示分章的书的目录表，其中每章都分节，而且每节都分小节。

27. 下面的饱和碳氢化合物有多少种不同的同分异构体？

a) C_3H_8　　b) C_5H_{12}　　c) C_6H_{14}

28. 在组织机构树里下述对象分别表示什么内容？

a) 一个顶点的父亲。

b) 一个顶点的儿子。

c) 一个顶点的兄弟。

d) 一个顶点的祖先。

e) 一个顶点的后代。

f) 一个顶点的层数。

g) 这个树的高度。

29. 对表示计算机文件系统的根树，回答与练习 28 所给的那些相同的问题。

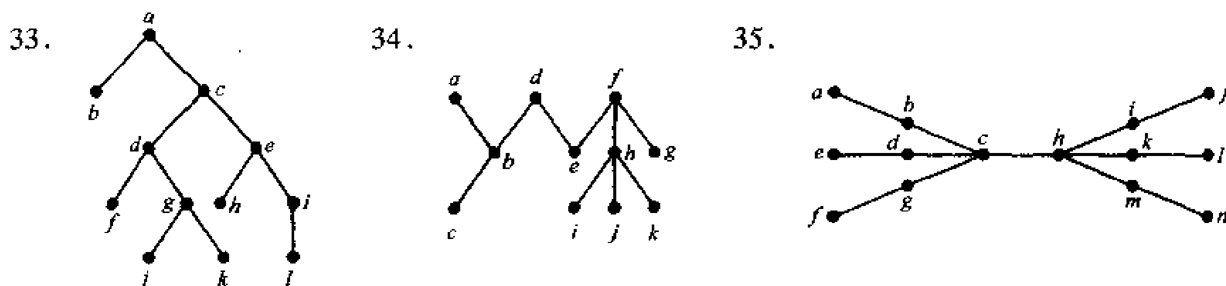
30. a) 画出表示 15 个处理器的树型连接网络的带有 15 个顶点的完全二叉树。

b) 说明如何用 a) 里的 15 个处理器分四步求 16 个数之和。

31. 设 n 是 2 的幂。证明：可以用 $n-1$ 个处理器的树型连接网络在 $\log n$ 步里求 n 个数之和。

*32. 标记树是其中每个顶点都指定了标记的树。当在两个标记树之间存在保持顶点标记的同构时，就把这两个标记树当作同构的。用集合 {1, 2, 3} 里三个不同的数来标记三个顶点的非同构标记树有多少种？用集合 {1, 2, 3, 4} 里四个不同的数来标记四个顶点的非同构标记树有多少种？

非根树里顶点的离心度是从这个顶点开始的最长的简单通路的长度。若在树里没有其他顶点比一个顶点的离心度更小，则这个顶点就称为中心。在练习 33～35 中，求每一个是所给树的中心的顶点。



36. 证明：为了从非根树产生高度最小的根树，就应当选择中心来作为根。

*37. 证明：树有一个中心或两个相邻的中心。

38. 证明：每一个树都可以用两种颜色来着色。

根斐波那契树 T_n 是以下面的方式来递归地定义的。T_1 和 T_2 都是包含单个顶点的根树，而对 $n=3, 4, \cdots$来说，根树 T_n 是以 T_{n-1}作为左子树和 T_{n-2}作为右子树的根来构造的。

39. 画出前七个根斐波那契树。

*40. 根斐波那契树 T_n 有多少个顶点、树叶和内点？其中 n 是正整数。它的高度是什么？

8.2 树的应用

8.2.1 引言

将要讨论可以用树来研究的三个问题。第一个问题是：应当如何对列表里的项进行排序，以便可以容易地找到项的位置？第二个问题是：为了在某种类型的一组对象里找出带某种性质的对象，应当做出一系列什么样的决策？第三个问题是：应当如何用位串来有效地编码一个字符集？

8.2.2 二叉搜索树

在列表里搜索一些项，是计算机科学所引起的最重要的任务之一。主要目标是实现一个搜索算法，当项都完全排序时，这个算法有效地找出项。这个任务可以通过使用**二叉搜索树**来完成，二叉搜索树是一种二叉树，其中任何顶点的每个儿子都指定为右儿子或左儿子，没有顶点具有一个以上的右儿子或左儿子，而且每个顶点都用一个关键字来标记，这个关键字是项中的一个。另外，这样指定顶点的关键字，使得顶点的关键字不仅大于它的左子树里的所有顶点的关键字，而且小于它的右子树里的所有顶点的关键字。

下面的递归过程用来形成项的列表的二叉搜索树。从只包含一个顶点（即根）的树开

始。指定列表中第一个项作为这个根的关键字。为了添加新的项，首先比较它与已经在树里的顶点的关键字，从根开始，若这个项小于所比较顶点的关键字而且这个顶点有左儿子，则向左移动，或者若这个项大于所比较顶点的关键字而且这个顶点有右儿子，则向右移动。当这个项小于所比较顶点的关键字而且这个顶点没有左儿子时，就插入以这个项作为关键字的一个新顶点来作为这个顶点的左儿子。同理，当这个项大于所比较顶点的关键字而且这个顶点没有右儿子时，就插入以这个项作为关键字的一个新顶点来作为这个顶点的右儿子。用下面的例子来说明这个过程。

例 1 构造下面这些单词的二叉搜索树（用字母顺序）：mathematics，physics，geography，zoology，meteorology，geology，psychology 和 chemistry。

解 8-16 显示了构造这个二叉搜索树所用的步骤。单词 mathematics 是根的关键字。因为 physics 是在 mathematics 之后（按照字母顺序），所以给根添加带关键字 physics 的右儿子。因为 geography 是在 mathematics 之前，所以给根添加带关键字 geography 的左儿子。下一步，给带关键字 physics 的顶点添加右儿子，并且给其指定关键字 zoology，因为 zoology 是在 mathematics 之后和在 physics 之后。同理，给带关键字 physics 的顶点添加左儿子，并且给其指定关键字 meteorology。给带关键字 geography 的顶点添加右儿子，并且给其指定关键字 geology。给带关键字 zoology 的顶点添加左儿子，并且给其指定关键字 psychology。给带关键字 geography 的顶点添加左儿子，并且给其指定关键字 chemistry。（读者应当完成在每步上所需的所有比较。） ■

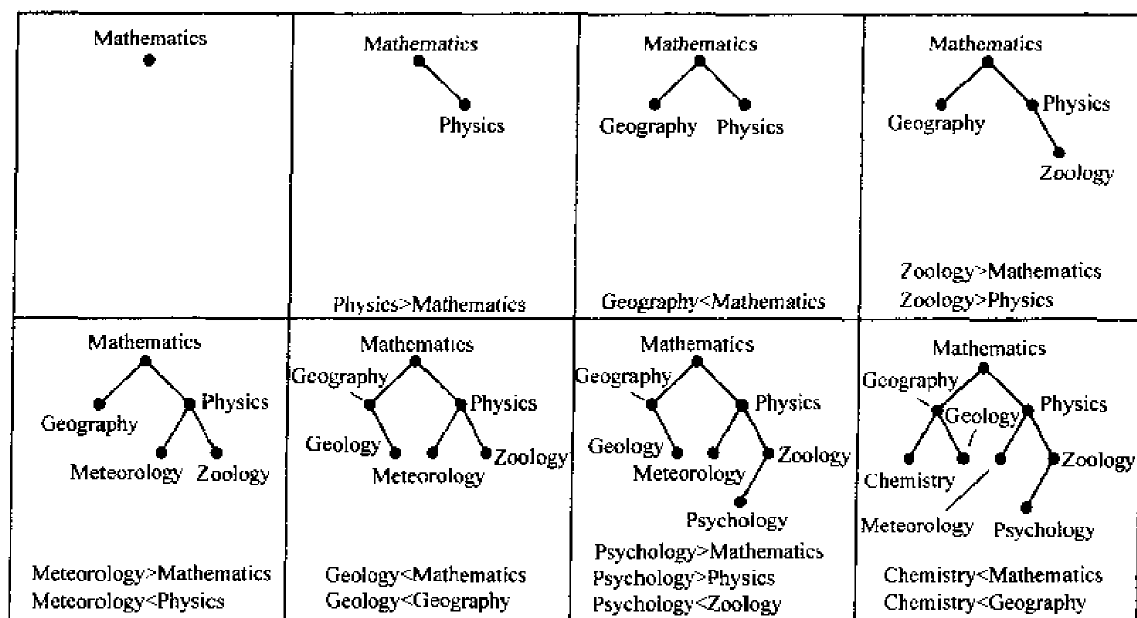


图 8-16 构造二叉搜索树

为了找出项的位置，试验添加它到二叉搜索树。若它出现，则找到它的位置。算法 1 给出这样的伪代码，它在二叉搜索树寻找一个项的位置，若没有找到这个项，则添加一个新的顶点，用这个项作为新顶点的关键字。当 x 不是关键字时，给树添加带关键字 x 的新顶点。在这个伪代码中，v 是以 x 作为其关键字的顶点，而 $label(v)$ 表示顶点 v 的关键字。

现在将确定这个过程的计算复杂性。假定有 n 个项的列表的二叉搜索树 T。可以从 T

这样构造一个满二叉树 U：在必要时添加无标记的顶点，以使得每个带关键字的顶点都有两个儿子。这个做法在图 8-17 里说明。一旦这样做了，就容易找出新项的位置，或者添加新项作为关键字而不添加顶点。

添加一个新项所需要的比较次数，最多是在 U 里从根到树叶的最长通路的长度。U 的内点都是 T 的顶点。所以 U 有 n 个内点。现在可以利用 8.1 节定理 4 的部分(ii)，得出 U 有 $n+1$ 个树叶。利用 8.1 节的推论 1，可以看出 U 的高度大于或等于 $h=\lceil\log(n+1)\rceil$。所以，必须至少执行 $\lceil\log(n+1)\rceil$ 次比较，以此添加某个项。注意，若 U 是平衡的，则它的高度是 $\lceil\log(n+1)\rceil$（根据 8.1 节的推论 1）。因此，若二叉搜索树是平衡的，则找出一个项的位置或者添加一个项只需要不超过 $\lceil\log(n+1)\rceil$ 次比较。当给二叉搜索树添加一些项时，该树可能变得不平衡。因为平衡的二叉搜索树给出二叉搜索的最优的最坏情形复杂性，所以已经设计出来了在添加项时重新平衡二叉搜索树的算法，感兴趣的读者可以查阅关于数据结构的参考文献来了解这些算法的描述。

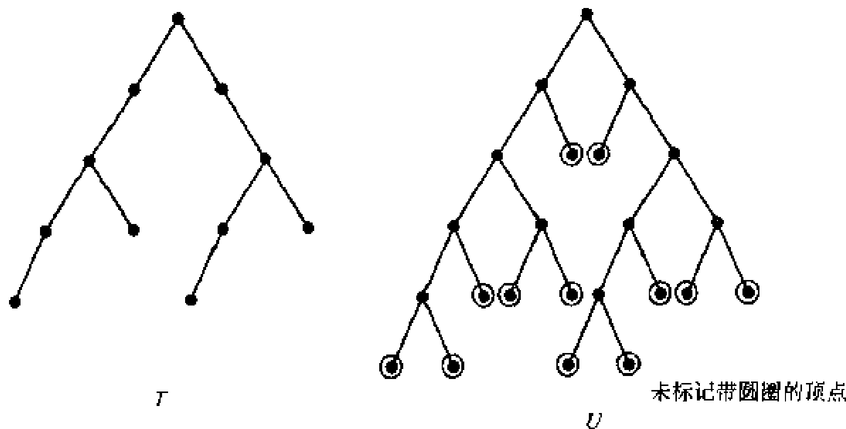


图 8-17　添加无标记顶点使得二叉搜索树成为满的

算法 1　二叉搜索树算法

```
procedure insertion(T:二叉搜索树, X:项)
v := T 的根
{一个不在 T 里出现的顶点具有值 null}
while v ≠ null 并且 label(v) ≠ x
begin
  if x < label(v) then
  if v 的左儿子 ≠ null then v := v 的左儿子
  else 添加 new vertex 作为 v 的左儿子并且设置 v := null
  else
  if v 的右儿子 ≠ null then v := v 的右儿子
  else 给 T 添加 new vertex 作为 v 的右儿子并且设置 v := null
end
if T 的根 = null then 给树添加顶点 r 并且用 x 标记它
else if label(v) ≠ x then 用 x 标记 new vertex
{v = x 的位置}
```

8.2.3　决策树

根树可以用来为这样的问题建立模型，其中一系列决策导致一个解。例如，二叉搜索树可以用来基于一系列比较来找出项的位置，其中每次比较都说明是否已经找到了项的位置，或者是否应当向右或向左进入子树。其中每个内点都对应着一次决策，这些顶点的子树都对应着该决策的每种可能后果，这样的根树称为决策树。问题的可能的解对应着这个根树的通向树叶的通路。下一个例子说明决策树的应用。

例 2　假定有重量相同的七枚硬币和重量较轻的一枚伪币。为了用一架天平秤确定这八枚硬币中哪个是伪币，需要多少次称重？给出找出这个伪币的算法。

解　在天平秤上每次称重结果有三种可能性。分别是：两个托盘有相同的重量，第一个托盘较重，或第二个托盘较重。所以，称重序列的决策树是 3 元树。在决策树里至少有八个树叶，这是因为有八种可能的后果（因为每枚硬币都可能是较轻的伪币），而每种可能的后果必须至少用一个树叶来表示。确定伪币所需要的最大称重次数是决策树的高度。从 8.1 节的推论 1 得出决策树的高度至少是 $\lceil \log_3 8 \rceil = 2$。因此，至少需要两次称重。

用两次称重来确定伪币，这是可行的。说明如何这样做的决策树如图 8-18 所示。■

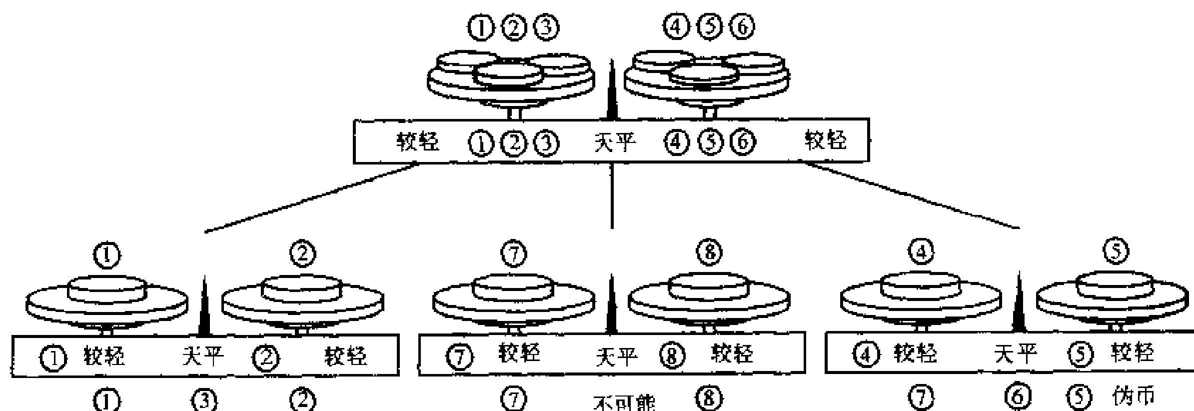


图 8-18　找出伪币位置的决策树

在本章第 4 节里将研究利用决策树的排序算法。

8.2.4　前缀码

考虑一下这样的问题：用位串来编码英语字母表示的字母（其中不区分小写和大写字母）。可以用长度为 5 的位串来表示每个字母，因为只有 26 个字母而有 32 个长度为 5 的位串。当每个字母都用 5 位来编码时，用来编码数据的总位数是 5 乘以文本中的字符数。有没有可能找出这些字母的编码方案，使得在编码数据时使用的位更少？若这样做是可以的，那么就可以节省存储空间而且缩短传输时间。

考虑用不同长度的位串来编码字母。较为频繁地出现的字母应当用较短的位串来编码，

较长的位串应当用来编码不经常出现的字母。当把字母编码成变化的位数时，就必须用某种方法来确定每个字母的位在何处开始和结束。例如，若把 *e* 编码成 0，把 *a* 编码成 1，而把 *t* 编码成 01，则位串 0101 可能对应着 *eat*、*tea*、*eaea* 或 *tt*。

为了保证没有位串对应着一个以上的字母序列，一个字母的位串永远不应当出现在另外一个字母的位串的开头部分。具有这个性质的编码称为**前缀码**。例如，把 *e* 编码成 0、把 *a* 编码成 10、而把 *t* 编码成 11 的编码就是前缀码。从编码一个单词的字母的唯一位串可以恢复这个单词。例如，串 10110 是 *ate* 的编码。为了看明白这一点，注意开始的 1 不表示一个字符，但是 10 表示 *a*（并且它不可能是另外一个字母的位串的开始部分）。然后，下一个 1 不表示一个字符，但是 11 表示 *t*。最后一位 0 表示 *e*。

前缀码可以用二叉树来表示，其中字符是树里树叶的标记。这样标记树的边，使得通向左儿子的边标记为 0 而通向右儿子的边标记为 1。用来编码一个字符的位串是在从根到以这个字符作为标记的的树叶的唯一通路上边的标记的序列。例如，图 8-19 里的树表示把 *e* 编码成 0，把 *a* 编码成 10，把 *t* 编码成 110，把 *n* 编码成 1110，和把 *s* 编码成 1111。

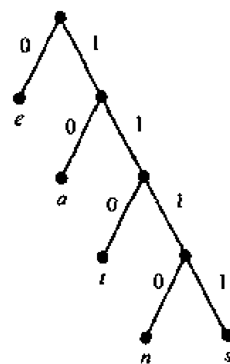


图 8-19　用前缀码表示的二叉树

表示编码的树可以用来解码位串。例如，考虑一个用图 8-19 里的编码来编码成 11111011100 的单词。这个位串可以这样解码：从根开始，用位的序列来形成一条到树叶为止的通路。每个 0 位都使得通路向下到达通向通路里上一个顶点的左儿子的边，而每个 1 位都对应到上一个顶点的右儿子。所以，开头的 1111 对应这样的通路：从根开始，向右前进四次，到达以 *s* 作为标记的树叶，因为串 1111 是 *s* 的编码。从第五个位继续进行，在向右再向左之后，就到达下一个树叶，这时访问以 *a* 作为标记的顶点，其中把 *a* 编码成 10。从第七个位开始，在向右三次然后向左之后，到达了下一个树叶，这时访问用 *n* 标记的顶点，其中把 *n* 编码成 1110。最后，末位 0 通向用 *e* 标记的树叶。因此，原来的单词是 *sane*。

可以从任何二叉树来构造一个前缀码，其中每个内点的左边都用 0 标记，而右边都用 1 标记，树叶都用字符标记。字符都用从根到这个树叶的唯一通路里的边的标记所组成的位串来编码。

存在一些比如霍夫曼算法这样的算法，它们可用来根据字符的出现频率来产生有效的编码。在这里将不给出这些算法的细节。（感兴趣的读者可以在本书末尾所给出的本节的参考文献里找到这些算法的细节）。

练习

1. 用字母顺序建立下面这些单词的二叉搜索树：*banana*，*peach*，*apple*，*pear*，*coconut*，*mango* 和 *papaya*。
2. 用字母顺序建立下面这些单词的二叉搜索树：*oenology*，*phrenology*，*campanology*，*ornithology*，*ichthyology*，*limnology*，*alchemy* 和 *astrology*。
3. 为了在练习 1 的搜索树里找出下面每个单词的位置或者添加它们，而且每次都重新开始，分别需要多少次比较？

   a) *pear*　　b) *banana*　　c) *kumquat*　　d) *orange*

4. 为了在练习 2 的搜索树里找出下面每个单词的位置或者添加它们，而且每次都重新开始，分别需要多少次比较？

   a) *palmistry*　　b) *etymology*　　c) *paleontolgy*　　d) *glaciology*

5. 用字母顺序构造下面句子里的单词的二叉搜索树：“*The quick brown fox jumps over the lazy dog*”。

6. 为了在 4 枚硬币中找出一枚较轻的伪币，需要多少次天平秤的称重？描述用这样次数的称重来找出较轻的伪币的算法。

7. 若一枚伪币比其他硬币既可能较重也可能较轻，那么为了在 4 枚硬币中找出这枚伪币，需要多少次天平秤的称重？描述用这样次数的称重来找出这枚伪币的算法。

*8. 若一枚伪币比其他硬币较重或较轻，那么为了在 8 枚硬币中找出这枚伪币，需要多少次天平秤的称重？描述用这样次数的称重来找出这枚伪币的算法。

*9. 若一枚伪币比其他硬币较轻，那么为了在 12 枚硬币中找出这枚伪币，需要多少次天平秤的称重？描述用这样次数的称重来找出这枚伪币的算法。

*10. 4 枚硬币中一枚可能是伪币。若它是伪币，则它比其他硬币既可能较重也可能较轻。那么为了确定是否有一枚伪币，以及若有伪币，它是比其他硬币较重还是较轻？使用一台天平秤，需要多少次称重？描述用这样次数的称重来找出这枚伪币并且确定它是较轻还是较重的算法。

11. 下面哪些编码是前缀码？

   a) a：11，e：00，t：10，s：01

   b) a：0，e：1，t：01，s：001

   c) a：101，e：11，t：001，s：011，n：010

   d) a：010，e：11，t：011，s：1011，n：1001，i：10101

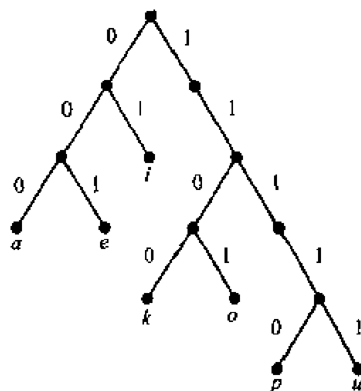
12. 构造表示下面编码方案的前缀码的二叉树。

   a) a：11，e：0，t：101，s：100

   b) a：1，e：01，t：001，s：0001，n：00001

   c) a：1010，e：0，t：11，s：1011，n：1001，i：10001

13. 若编码方案是用右边的树来表示，那么什么是 a，e，i，k，o，p 和 u 的编码？



14. 给定编码方案 a：001，b：0001，e：1，r：0000，s：0100，t：011，x：01010，找出用下面的位串来表示的单词。

   a) 01110100011　　b) 0001110000

   c) 0100101010　　d) 01100101010

8.3 树的遍历

8.3.1 引言

有序根树常常用来保存信息。需要一些访问有序根树的每个顶点以存取数据的算法。将描述几个重要的访问有序根树的所有顶点的算法。有序根树也可以用来表示各种类型的表达式，比如由数字、变量和运算所组成的算术表达式。对用来表示这些表达式的有序

根树来说，它的顶点的一些不同的列表在这些表达式的求值中是有用处的。

8.3.2 通用地址系统

遍历有序根树的所有顶点的过程，都依赖于儿子的顺序。在有序根树里，一个内点的儿子从左向右地显示在表示这些有向图的图形里。

将描述一种完全地排序有序根树顶点的方法。为了产生这个顺序，必须首先标记所有的顶点。如下递归地完成这件事。

1. 用整数 0 标记根。然后用 1，2，3，…，k 从左向右标记它的 k 个儿子（在 1 层上）。
2. 对在 n 层上带标记 A 的每个顶点 v，按照从左向右画出它的 k_v 个儿子的顺序，用 $A.1$，$A.2$…，$A.k_v$ 标记它的 k_v 个儿子。

遵循这个过程，对 $n \geqslant 1$ 来说，在 n 层上的顶点 v 标记成 $x_1.x_2.\cdots x_n$，其中从根到 v 的唯一通路经过 1 层的第 x_1 个顶点，经过 2 层的第 x_2 个顶点，依次类推。这样的标记称为根树的**通用地址系统**。

可以利用顶点在通用地址系统里的标记的字典顺序来完全地排序这些顶点。若存在 i（$0 \leqslant i \leqslant n$）满足 $x_1 = y_1$，$x_2 = y_2$，…，$x_{i-1} = y_{i-1}$，并且 $x_i < y_i$；或者若 $n < m$ 并且对 $i = 1, 2, \cdots, n$ 来说 $x_i = y_i$，那么标记着 $x_1.x_2.\cdots x_n$ 的顶点就小于标记着 $y_1.y_2.\cdots y_m$ 的顶点。

例 1 在如图 8-20 所示的有序根树的顶点的旁边，显示出通用地址系统的标记。这些标记的字典顺序是

$$0<1<1.1<1.2<1.3<2<3<3.1<3.1.1<3.1.2<3.2.2.1<3.1.2.2$$
$$<3.1.2.3<3.1.2.4<3.1.3<3.2<4<4.1<5<5.1<5.1.1<5.2<5.3$$ ■

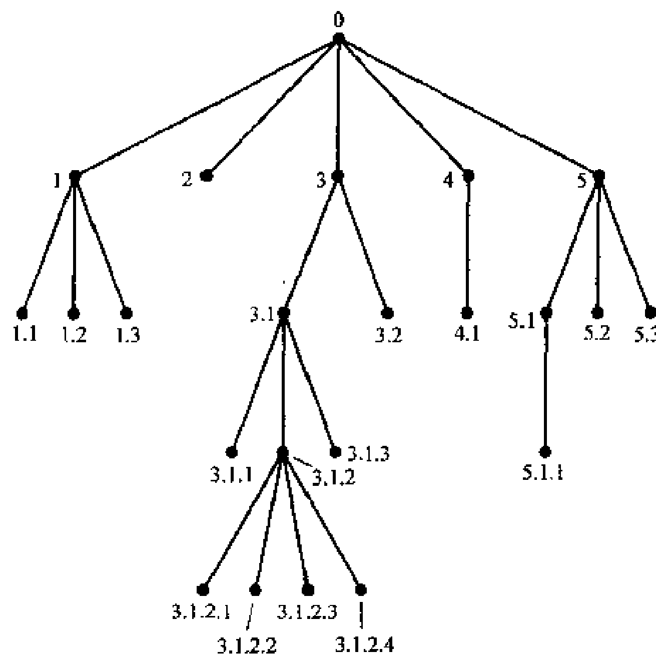


图 8-20 有序根树的通用地址系统

8.3.3 遍历算法

系统地访问有序根树每个顶点的过程都称为**遍历算法**。将描述三个最常用的这种算法：

前序遍历、中序遍历和后序遍历。这些算法每个都可以递归地定义。首先定义前序遍历。

定义 1 设 T 是带根 r 的有序根树。若 T 只包含 r，则 r 是 T 的前序遍历。否则，假定 T_1，T_2，⋯，T_n 是 T 里在 r 处从左向右的子树。前序遍历首先访问 r。它接着前序遍历 T_1，然后前序遍历 T_2，依次类推，直到前序遍历了 T_n 为止。

读者应当验证，有序根树的前序遍历给出与利用通用地址系统所得出的顺序相同的顶点顺序。图 8-21 说明如何执行前序遍历。

下面的例子说明前序遍历。

例 2 前序遍历以什么顺序访问图 8-22 所示的有序根树里的顶点?

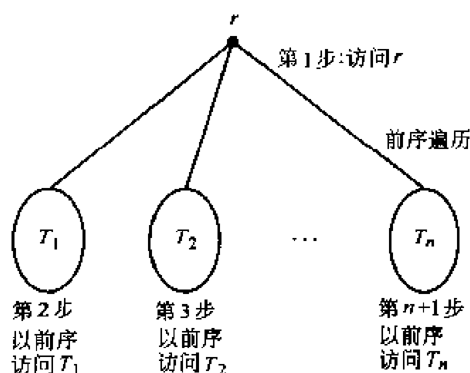


图 8-21 前序遍历

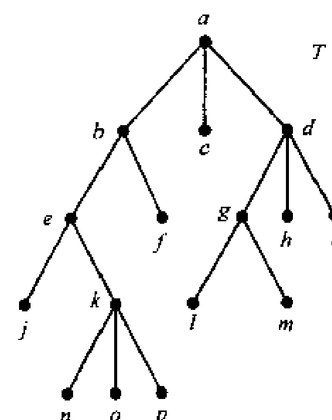


图 8-22 有序根树 T

解 T 的前序遍历的步骤显示在图 8-23 里。这样以前序来遍历 T，首先列出根 a，接着依次是带根 b 的子树的前序列表，带根 c 的子树（它只有 c）的前序列表，和带根 d 的子树的前序列表。

带根 b 的子树的前序列表首先列出 b，再以前序列出带根 e 的子树的顶点，然后以前序列出带根 f 的子树（它只有 f）的顶点。带根 d 的子树的前序列表首先列出 d，接着是带根 g 的子树的前序列表，接着是带根 h 的子树（它只有 h），接着是带根 i 的子树（它只有 i）。

带根 e 的子树的前序列表首先列出 e，接着是带根 j 的子树（它只有 j）的前序列表，接着是带根 k 的子树的前序列表。带根 g 的子树的前序列表是 g 接着 l，接着是 m。带根 k 的子树的前序列表是 k，n，o，p。所以，T 的前序遍历是 a，b，e，j，k，n，o，p，f，c，d，g，l，m，h，i。 ■

现在定义中序遍历。

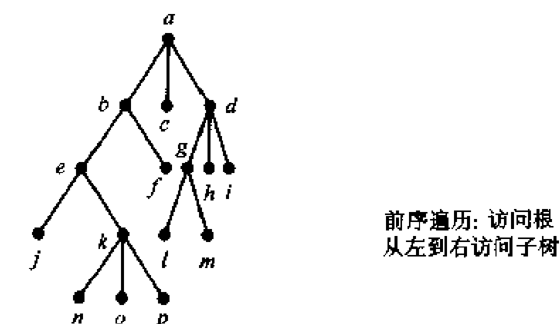
定义 2 设 T 是带根 r 的有序根树。若 T 只包含 r，则 r 是 T 的中序遍历。否则，假定 T_1，T_2，⋯，T_n 是 T 里在 r 处从左向右的子树。中序遍历首先以前序遍历 T_1，然后访问 r。它接着中序遍历 T_2，依次类推，直到中序遍历了 T_n 为止。

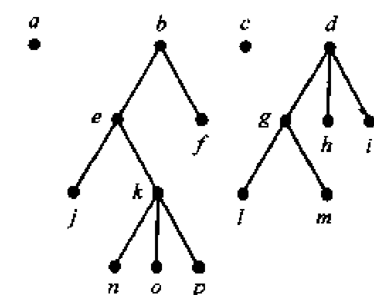
图 8-24 说明如何执行中序遍历。

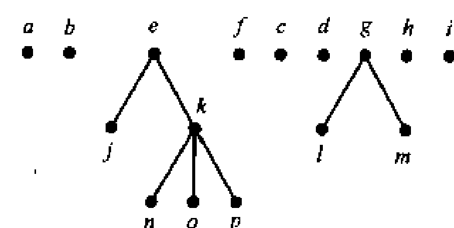
下面的例子说明中序遍历是如何完成的。

例 3 中序遍历以什么顺序访问图 8-22 所示的有序根树里的顶点?

解 T 的中序遍历的步骤显示在图 8-25 里。中序遍历首先是带根 b 的子树的中序遍历，然后是根 a，带根 c 的子树（它只有 c）的中序列表，和带根 d 的子树的中序列表。







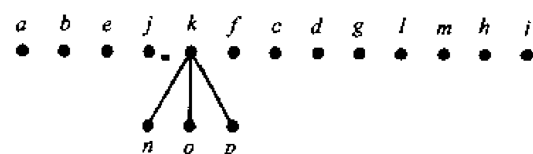




图 8-23　T 的前序遍历

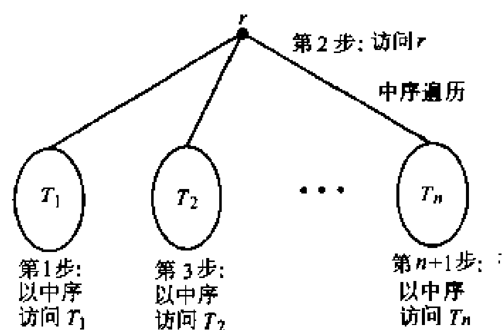
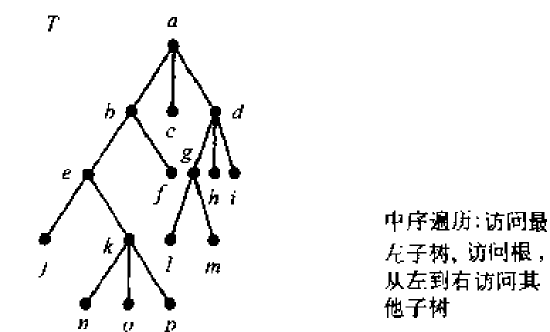


图 8-24　中序遍历

带根 b 的子树的中序列表，首先是带根 e 的子树的中序列表，然后是根 b，以及 f。带根 d



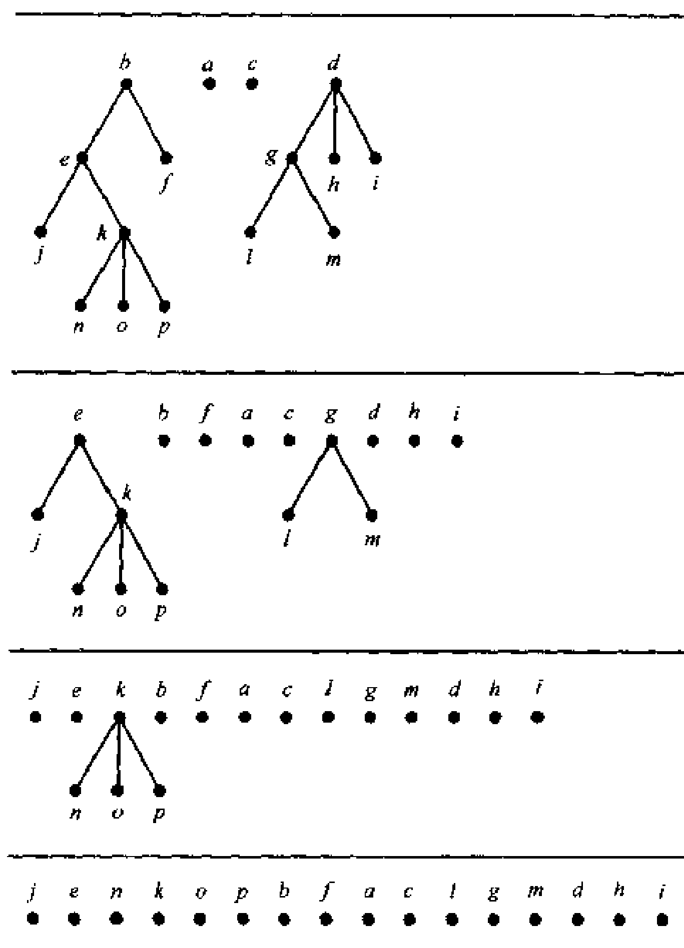


图 8-25　T 的中序遍历

的子树的中序列表，首先是带根 g 的子树的中序列表，接着是根 d，接着是 h，接着是 i。

带根 e 的子树的中序列表 j，接着是根 e，接着是带根 k 的子树的中序列表。带根 g 的子树的中序列表是 l，g，m。带根 k 的子树的中序列表是 n，k，o，p。所以，这个根树的中序遍历是 j，e，n，k，o，p，b，f，a，c，l，g，m，d，h，i。■

下面是后序遍历的定义。

定义 3　设 T 是带根 r 的有序根树。若 T 只包含 r，则 r 是 T 的后序遍历。否则，假定 T_1，T_2，…，T_n 是 T 里在 r 处从左向右的子树。后序遍历首先后序遍历 T_1，然后后序遍历 T_2，…，然后后序遍历 T_n，最后访问 r。

图 8-26 说明后序遍历是如何执行的。下面的例子说明后序遍历如何工作。

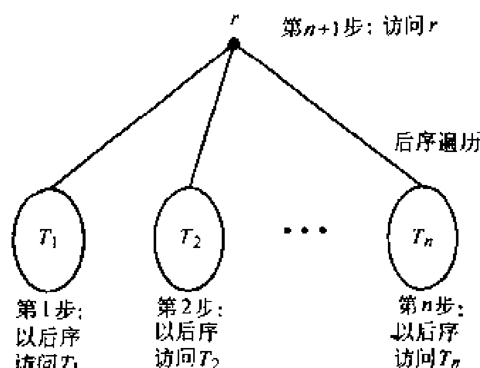


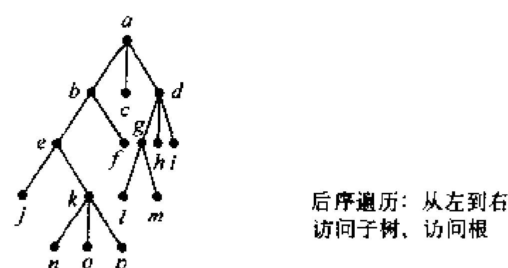
图 8-26　后序遍历

例 4　*后序遍历以什么顺序访问图 8-22 所示有序根树里的顶点?*

解　有序根树 T 的中序遍历的步骤显示在图 8-27 中。后序遍历首先是带根 b 的子树的后序遍历，然后是带根 c 的子树（它只有 c）的后序遍历，带根 d 的子树的后序遍历，接着是根 a。

带根 b 的子树的后序遍历首先是带根 e 的子树的后序遍历，接着是 f，接着是根 b。带根 d 的子树的后序遍历首先是带根 g 的子树的后序遍历，接着是 h，接着是 i，接着是根 d。

带根 e 的子树的后序遍历是 j，接着是带根 k 的子树的后序遍历，接着是根 e。带根 g 的子树的后序遍历是 l，m，g。带根 k 的子树的后序遍历是 n，o，p，k。因此，根树 T 的中序遍历是 j，n，o，p，k，e，f，b，c，l，m，g，h，i，d，a。■



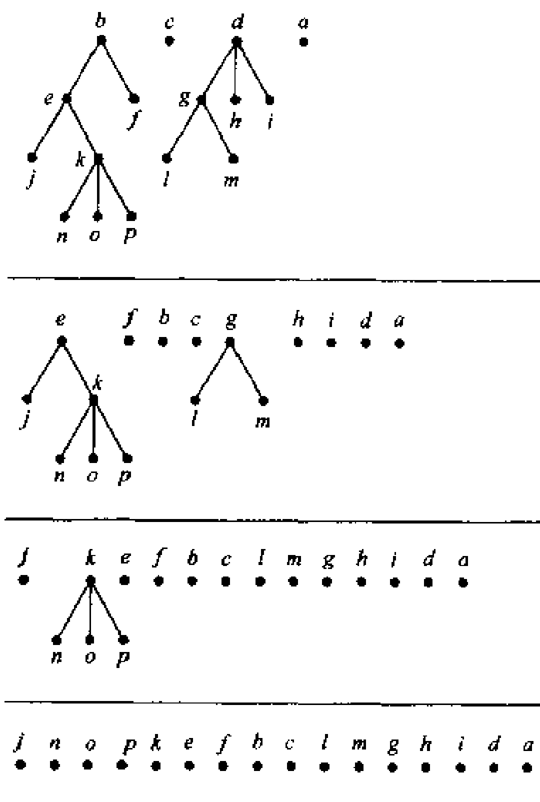


图 8-27　T 的后序遍历

存在一些容易的方法以前序、中序和后序来列出有序根树的顶点。为了这样做，首先从根开始围绕有序根树画一条曲线，如图 8-28 中的例子所示，沿着边移动。可以这样按照前序来列出顶点：当曲线第一次经过一个顶点时，就列出这个顶点。可以这样按照中序来列出顶点：当曲线第一次经过一个树叶时，就列出这个树叶，当曲线第二次经过一个内点时就列出这个内点。可以这样按照后序来列出顶点：当曲线最后一次经过一个顶点而返回这个顶点的父亲时，就列出这个顶点。当在图 8-28 的根树里这样做时，结果是前序遍历给出 a，b，d，h，e，i，j，c，f，g，k，中序遍历给出 h，d，b，i，e，j，a，f，c，k，g，后序遍历给出 h，d，i，j，e，b，f，k，g，c，a。

这些以前序、中序和后序遍历有序根树的算法，最容易用递归来表示。

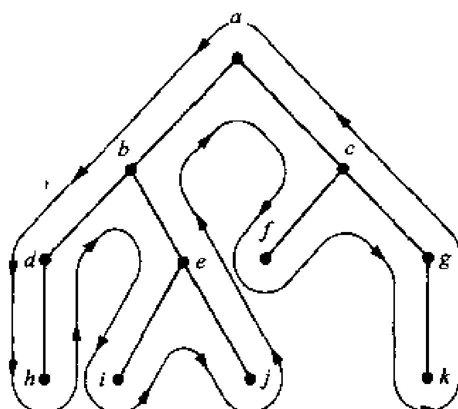


图 8-28　以前序、中序和后序遍历有序根树的快捷方法

```
算法 1　前序遍历
procedure preorder（T：有序根树）
r := T 的根
列出 r
for 从左到右的 r 的每个儿子 c
begin
   T(c):=以 c 为根的子树
   preorder(T(c))
end
```

```
算法 2　中序遍历
procedure inorder(T:有序根树)
r := T 的根
if r 是树叶 then 列出 r
else
begin
   l :=从左到右的 r 的第一个儿子
   T(l):=以 l 为根的子树
   inorder(T(l))
   列出 r
   for 除 l 外从左到右的 r 的每个儿子 c
      T(c):=以 c 为根的子树
      inorder(T(c))
end
```

算法 3 后序遍历

```
procedure postorder(T:有序根树)
r := T 的根
for 从左到右的 r 的每个儿子 c
begin
    T(c):=以 c 为根的子树
    postorder (T(c))
end
列出 r
```

8.3.4 中缀、前缀和后缀记法

可以用有序树来表示复杂的表达式，比如复合命题，集合的组合，以及算术表达式。例如，考虑由运算 +（加）、-（减）、*（乘）、/（除）和 ↑（幂）所组成的算术表达式的表示。将用括号来指出运算次序。有序根树可以用来表示这样的表达式，其中内点表示运算，树叶表示变量或数字。每个运算都作用在它的左子树和右子树上（以这个顺序）。

例 5 表示表达式$((x+y)\uparrow 2)+((x-4)/3)$的有序根树是什么？

解 这个表达式的二叉树可以自底向上来构造。首先，构造表达式 $x+y$ 的子树。然后把这个子树作为表示$(x+y)\uparrow 2$的更大子树的一部分。同样，构造表达式 $x-4$ 的子树，然后加入这个子树到表示$(x-4)/3$ 的子树里。最后组合表示$(x+y)\uparrow 2$ 和$(x-4)/3$ 的子树，形成表示$((x+y)\uparrow 2))+((x-4)/3)$的有序根树。这些步骤显示在图 8-29 里。 ■

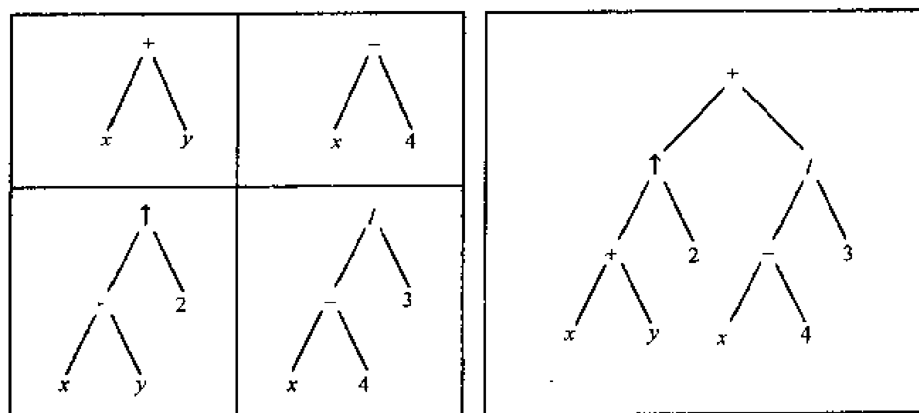


图 8-29 表示$((x+y)\uparrow 2)+((x-4)/3)$的二叉树

对表示一个表达式的二叉树的中序遍历，产生原来的表达式，其中元素和运算都是按它们原来的出现次序，例外的是一元运算，它们紧随运算对象。例如，图 8-30 里的二叉树分别表示表达式$(x+y)/(x+3)$，$(x+(y/x))+3$ 和 $x+(y/(x+3))$，对它们的中序遍历都得出中缀表达式 $x+y/x+3$。为了让这样的表达式无二义性，每当遇到运算时，就有必要在中序遍历

里包含括号。以这种方式获得的带完整括号的表达式称为是中缀形式。

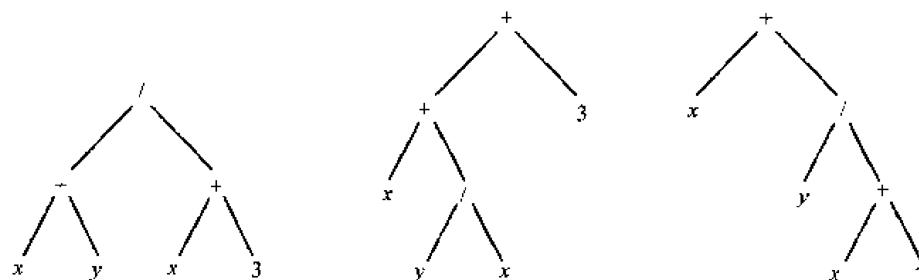


图 8-30 表示$(x+y)/(x+3)$，$(x+(y/x))+3$ 和 $x+(y/(x+3))$ 的根树

当以前序遍历表达式的根树时，就获得它的前缀形式。写成前缀形式的表达式称为是波兰记法，它是根据逻辑学家扬·卢卡锡维茨[1]来命名的（其实他是乌克兰人而非波兰人）。前缀记法下的表达式（其中每个运算都有规定的运算对象数）是无二义性的。对这个事实的验证就留给读者作为练习。

例 6 $((x+y)\uparrow 2)+((x-4)/3)$ 的前缀形式是什么？

解 通过遍历图 8-29 所示的表示这个表达式的二叉树，就获得它的前缀形式。这样就产生 + ↑ + x y 2 / − x 4 3。■

在表达式的前缀形式里，二元运算符（比如 +）在两个运算对象之前。因此，可以从右向左地求前缀形式的表达式的值。当遇到一个运算符时，就对在这个运算右边紧接着的两个运算对象执行相应的运算。另外，每当一个运算执行时，就认为结果是新的运算对象。

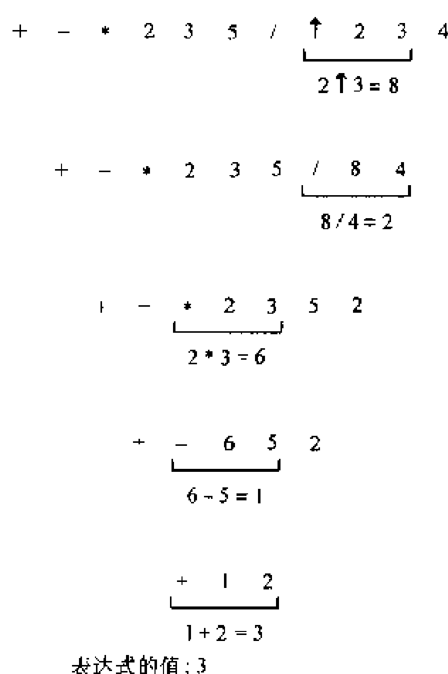


图 8-31 求一个前缀表达式的值

例 7 前缀表达式 + − * 2 3 5 / ↑ 2 3 4 的值是什么？

解 从右向左地求这个表达式的值所用的步骤，以及用右边的运算对象执行的运算，如图 8-31 所示。这个表达式的值是 3。■

通过后序遍历表达式的二叉树，就获得它的后缀形式。写成后缀形式的表达式称为是逆波兰记法。逆波兰记法下的表达式是无二义性的，所以不需要括号。对这个事实的验证留给读者。

[1] 扬·卢卡锡维茨(Jan Lukasiewicz,1878—1956) 卢卡锡维茨出生在勒沃，在勒沃大学学习并开始职业生涯。后来他转到华沙任教授职务。在第二次世界大战后，他被任命担任在都柏林的皇家爱尔兰学院的职务。卢卡锡维茨工作在多值逻辑的领域里；他在 1921 年关于三值逻辑的论文是对这个题目的重要贡献。不过，在数学界他最著名的是因为他引入了无括号记法，现在称为波兰记法。

例8 $((x+y)\uparrow 2)+((x-4)/3)$的后缀形式是什么？

解 这个表达式的后缀形式是这样获得的：执行图8-29所示的表示它的二叉树的后序遍历，这样就产生后缀表达式 $x\ y\ +\ 2\ \uparrow\ x\ 4\ -\ 3\ /\ +$。 ■

在表达式的后缀形式里，二元运算是在它的两个运算对象之后。所以，为了从一个表达式的后缀形式来求它的值，就从左向右进行，每当一个运算符跟在两个运算对象后面时，就执行这个运算。在一个运算执行之后，这个运算的结果就成为一个新的运算对象。

例9 后缀表达式 $7\ 2\ 3\ *\ -\ 4\ \uparrow\ 9\ 3\ /\ +$ 的值是什么？

解 如图8-32所示，求这个表达式的值所用的步骤是这样的：从左边开始，当两个运算对象后面接着一个运算符时，就执行这个运算。这个表达式的值是4。 ■

7 2 3 * − 4 ↑ 9 3 / +

$2*3=6$

7 6 − 4 ↑ 9 3 / +

$7-6=1$

1 4 ↑ 9 3 / +

$1^4=1$

1 9 3 / +

$9/3=3$

1 3 +

$1+3=4$

图8-32 求一个后缀表达式的值

根树可以用来表示其他类型的表达式，比如那些表示复合命题和集合组合的表达式。在这些例子里出现比如命题否定这样的一元远算。为了表示这样的运算符及其运算对象，就用顶点表示运算符并且用这个顶点的儿子表示运算对象。

例10 求表示复合命题$(\neg(p\wedge q))\leftrightarrow(\neg p\vee\neg q)$的有序根树。然后用这个根树求这个表达式的前缀、后缀和中缀形式。

解 这个复合命题的有序根树是自底向上地构造的。首先，构造$\neg p$和$\neg q$的子树（其中把¬当作一元运算符）。另外，构造$p\wedge q$的子树。然后构造$\neg(p\wedge q)$和$(\neg p)\vee(\neg q)$的子树。最后，用这两个子树来构造最终的根树。这个过程的步骤显示在图8-33里。

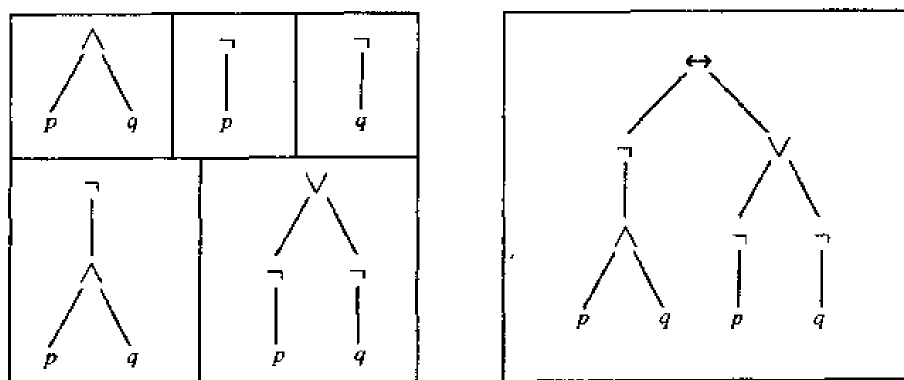


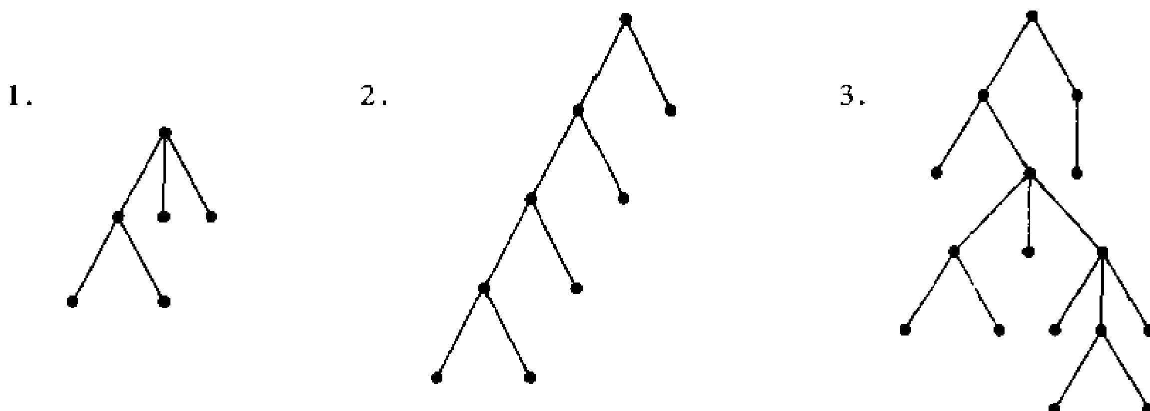
图8-33 构造一个复合命题的根树

这个表达式的前缀、后缀和中缀形式是这样求出的：分别以前序、后序和中序遍历这个根树（包含括号）。这些遍历分别给出$\leftrightarrow\neg\wedge pq\vee\neg p\ \neg q$，$pq\wedge\neg p\ \neg q\ \neg\vee\leftrightarrow$和$(\neg(p\wedge q))\leftrightarrow((\neg p)\vee(\neg q))$。 ■

因为前缀表达式和后缀表达式都是无二义性的，而且因为不用来回扫描就容易求出它们的值，所以它们在计算机科学里大量使用。这样的表达式对编译器的构造是特别有用的。

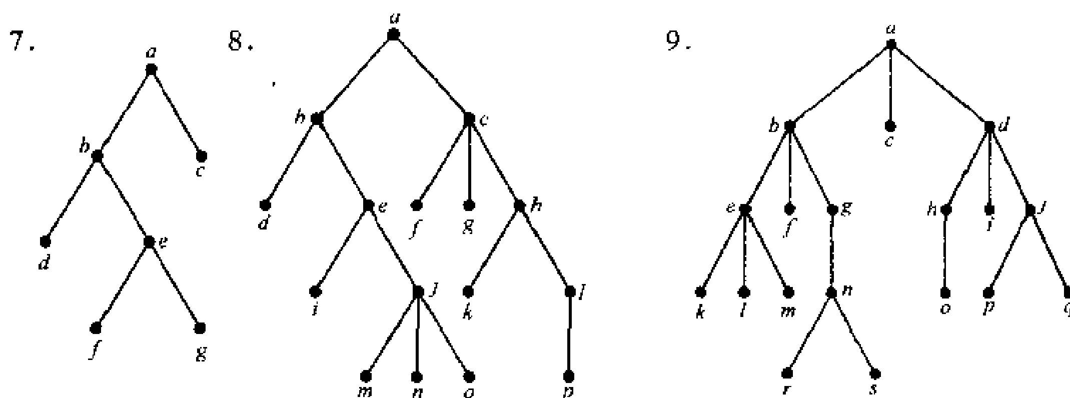
练习

在练习 1～3 中，对给定的有序根树来构造通用地址系统。然后利用这个通用地址系统来排序使用顶点标记的字典顺序的顶点。



4. 假定在有序根树 T 里顶点 v 的地址是 3.4.5.2.4。
   a) v 是在哪一层？
   b) v 的父亲的地址是什么？
   c) v 的兄弟最少有多少？
   d) 若 v 具有这个地址，那么在 T 里最少可能有多少个顶点？
   e) 求其他必然出现的地址。
5. 假定在有序根树 T 里地址最大的顶点的地址是 2.3.4.3.1。是否有可能确定 T 里的顶点数？
6. 有序根树的树叶能否具有下面的通用地址表？若能，则构造出这样的有序根树。
   a) 1.1.1，1.1.2，1.2，2.1.1.1，2.1.2，2.1.3，2.2，3.1.1，3.1.2.1，3.1.2.2，3.2
   b) 1.1，1.2.1，1.2.2，1.2.3，2.1，2.2.1，2.3.1，2.3.2，2.4.2.1，2.4.2.2，3.1，3.2.1，3.2.2
   c) 1.1，1.2.1，1.2.2，1.2.2.1，1.3，1.4，2，3.1，3.2，4.1.1.1

在练习 7～9 中，确定前序遍历访问所给的有序根树的顶点的序列。



10. 使用中序遍历，以什么顺序访问练习7中有序根树的顶点？
11. 使用中序遍历，以什么顺序访问练习8中有序根树的顶点？
12. 使用中序遍历，以什么顺序访问练习9中有序根树的顶点？
13. 使用中序遍历，以什么顺序访问练习7中有序根树的顶点？
14. 使用中序遍历，以什么顺序访问练习8中有序根树的顶点？
15. 使用后序遍历，以什么顺序访问练习9中有序根树的顶点？
16. 用二叉树来表示表达式$((x+2)\uparrow 3)*(y-(3+x))-5$。
17. 把练习16中的表达式写成
    a) 前缀记法　b) 后缀记法　c) 中缀记法
18. 用二叉树来表示表达式$(x+xy)+(x/y)$和$x+((xy+x)/y)$。
19. 把练习18中的表达式写成
    a) 前缀记法　b) 后缀记法　c) 中缀记法
20. 用有序根树来表示复合命题$\neg(p\wedge q)\leftrightarrow(\neg p\vee\neg q)$和$(\neg p\wedge(q\leftrightarrow\neg p))\vee\neg q$。
21. 把练习20中的表达式写成
    a) 前缀记法　b) 后缀记法　c) 中缀记法
22. 用有序根树来表示$(A\cap B)-(A\cup(B-A))$。
23. 把练习22中的表达式写成
    a) 前缀记法　b) 后缀记法　c) 中缀记法

*24. 能够用多少种方式给字符串$\neg p\wedge q\leftrightarrow\neg p\vee\neg q$完全加上括号以便产生中缀表达式？

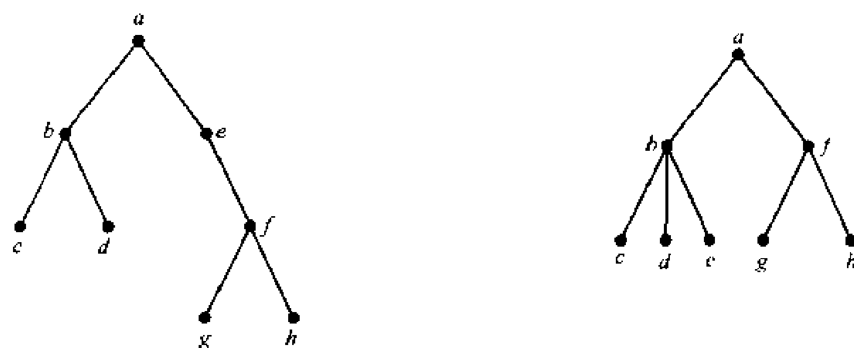
*25. 能够用多少种方式给字符串$A\cap B-A\cup B-A$完全加上括号以便产生中缀表达式？

26. 画出下面用前缀记法所写的每个算术表达式所对应的有序根树。然后用中缀记法来写每个表达式。
    a) + * + - 5 3 2 1 4　b) ↑ + 2 3 - 5 1　c) * / 9 3 + * 2 4 - 7 6
27. 下面每个前缀表达式的值是什么？
    a) - * 2 / 8 4 3　b) ↑ - * 3 3 * 4 2 5
    c) + - ↑ 3 2 + ↑ 2 3 / 6 - 4 2　d) * + 3 + 3 ↑ 3 + 3 3 3
28. 下面每个后缀表达式的值是什么？
    a) 5 2 1 - - 3 1 4 + + *　b) 9 3 / 5 + 7 2 - *　c) 3 2 * 2 ↑ 5 3 - 8 4 / * -
29. 构造前序遍历为$a, b, f, c, g, h, i, d, e, j, k, l$的有序根树，其中$a$有4个儿子，$c$有3个儿子，$j$有2个儿子，$b$和$e$都有1个儿子，所有其他顶点都是树叶。

*30. 证明：当规定了有序根树的前序遍历所生成的顶点列表，并且规定了每个顶点的儿子数时，这个有序根树是唯一确定的。

*31. 证明：当规定了有序根树的后序遍历所生成的顶点列表，并且规定了每个顶点的儿子数时，这个有序根树是唯一确定的。

32. 证明：下面所示的两个有序根树的前序遍历产生相同的顶点列表。注意这个结果不与练习30中的命题相矛盾，因为在这两个有序根树里内点的儿子数是不同的。



33．证明：下面所示的两个有序根树的后序遍历产生相同的顶点列表。注意这个结果不与练习 31 中的命题相矛盾，因为在这两个有序根树里内点的儿子数是不同的。



在一组符号和一组二元运算符上用前缀记法表示的合式公式是用下面的规则来递归地定义的：

(i) 若 x 是符号，则 x 是用前缀记法表示的合式公式；

(ii) 若 X 和 Y 都是合式公式并且 $*$ 是运算符，则 $*XY$ 是合式公式。

34．下列哪些公式是在符号 $\{x, y, z\}$ 和二元运算符集 $\{\times, +, \circ\}$ 上的合式公式？

a) $\times + + x y z$　b) $\circ x y \times x z$　c) $\times \circ x z \times \times x y$　d) $\times + \circ x x \circ x x x$

*35．证明：在一组符号和一组二元运算符上用前缀记法表示的任何合式公式所包含的符号数都比运算符数恰好多一个。

36．给出在一组符号和一组二元运算符上用后缀记法表示的合式公式的定义。

37．给出在符号集 $\{x, y, z\}$ 和二元运算符集 $\{\times, +, \circ\}$ 上带 3 个或 3 个以上运算符的用后缀记法表示的合式公式的 6 个例子。

38．把用前缀记法表示的合式公式的定认推广到这样的符号集和运算符集上，其中运算符可能不是二元的。

8.4　树与排序

8.4.1　引言

对一个集合里的元素排序的问题出现在许多场合里。例如，为了产生打印的电话目录，就有必要以字母顺序来排列用户姓名。

假定存在着一个集合里的元素之间的全序。起初一个集合里的元素可能处在任意顺序里。排序就是重新排列这些元素形成一个列表，其中元素都以升序排列。例如，对列表 7，2，1，4，5，9 的排序就产生出列表 1，2，4，5，7，9。对列表 d，h，c，a，f 的

排序（利用字母序）就产生出列表 a，c，d，f，h。

计算机使用中占有很大百分比的是用来排序各种各样的东西。因此，已经投入很多努力来发展有效的排序算法。在本节里，将讨论几个排序算法和它们的计算复杂性。在本节里将看到用树来描述排序算法，并且用在这些算法的复杂性分析里。

8.4.2　排序的复杂性

已经发展了许多不同的排序算法。为了确定一个具体的排序算法是否有效，要确定这个算法的复杂性。用树作为模型，可以求出排序算法的最坏情形复杂性的下界。

n 个元素有 $n!$ 种可能的排序，这是因为这些元素的 $n!$ 种排列每一个都可以是正确的顺序。将研究的排序算法都是基于二叉比较，即一次比较两个元素。每次这样的比较都缩小了可能的排序集合。因此，基于二叉比较的排序算法可以表示成二叉决策树，其中每个内点表示两个元素的一次比较。每个树叶表示 n 个元素的 $n!$ 种排列中的一种。

例 1　在图 8-34 里显示排序列表 a，b，c 里元素的决策树。■

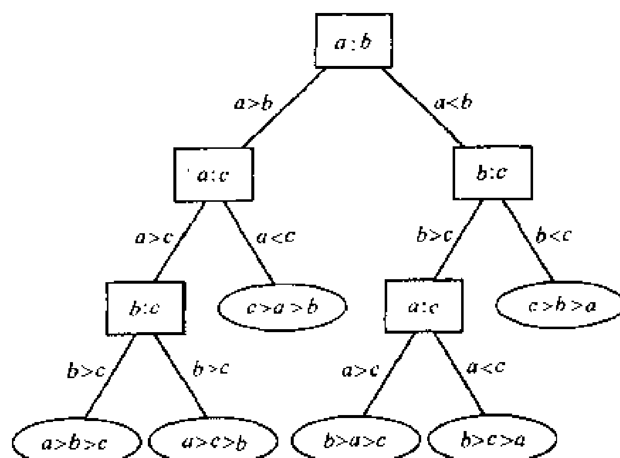


图 8-34　排序三个不同元素的决策树

对基于二叉比较的排序来说，其复杂性是通过所用的二叉比较的次数来度量的。排序有 n 个元素的列表所需要的最多比较次数就给出了这个算法的最坏情形复杂性。所用的最多比较次数等于表示这个排序过程的决策树中最长路径的长度。换句话说，所需要的最多比较次数等于这个决策树的高度。因为带 $n!$ 个树叶的二叉树的高度至少是 $\lceil \log n! \rceil$（利用 8.1 节推论 1），所以下面定理 1 表明至少需要 $\lceil \log n! \rceil$ 次比较。

定理 1　基于二叉比较的排序算法至少需要 $\lceil \log n! \rceil$ 次比较。

根据 8.1 节例 5，得出 $\lceil \log n! \rceil$ 是 $O(n\log n)$。事实上，对 $n>4$ 来说它大于 $n\log n/4$（见练习 18），所以用比较作为排序手段的排序算法都没有优于 $O(n\log n)$ 的最坏情形时间复杂性。因此，若一个排序算法具有 $O(n\log n)$ 的时间复杂性，则它就是尽可能地有效的（在时间复杂性的大 O 估计的意义下）。

8.4.3　冒泡排序

冒泡排序是最简单的排序算法之一，但不是最有效的排序算法之一。它把一个列表这样排列成升序：相继地比较相邻的元素，若它们顺序不对，则交换它们。为了完成冒

泡排序，从表头开始执行基本操作，即交换一个较大元素与它后面的较小元素，对整个列表完全执行一遍。让这个过程迭代，直到排序宣告完成为止。可以想象把表里的元素排成垂直一列。在冒泡排序里，当交换较小的元素与较大的元素时，较小的元素就“冒泡”到顶上。较大的元素则“下沉”到底下。在下面的例子里对此进行说明。

例 2　用冒泡排序把 3，2，4，1，5 排列成升序。

解　首先比较前两个元素 3 和 2。因为 $3>2$，所以交换 3 与 2，产生列表 2，3，4，1，5。因为 $3<4$，所以继续比较 4 和 1。因为 $4>1$，所以交换 4 与 1，产生列表 2，3，1，4，5。因为 $4<5$，所以第 1 遍就完成了。第 1 遍保证最大元素 5 是在正确位置上。

第 2 遍首先比较 2 和 3。因为这两个数是在正确顺序里，所以比较 3 和 1。因为 $3>1$，所以交换这两个数，产生 2，1，3，4，5。因为 $3<4$，所以这两个数是在正确顺序里。对这一遍来说没有必要去做更多的比较，因为 5 已经是在正确位置上。第 2 遍保证两个最大元素 4 和 5 都是在正确位置上。

第 3 遍首先比较 2 和 1。因为 $2>1$，所以交换这两个数，产生 1，2，3，4，5。因为 $2<3$，所以这两个数是在正确顺序里。对这一遍来说没有必要去做更多的比较，因为 4 和 5 都已经在正确位置上。第 3 遍保证三个最大元素 3，4，和 5 都是在正确位置上。

第 4 遍包括一次比较，即 1 和 2 的比较。因为 $1<2$，所以这两个数是在正确顺序里。这样就完成了冒泡排序。

这个算法的步骤在图 8-35 里说明。　■

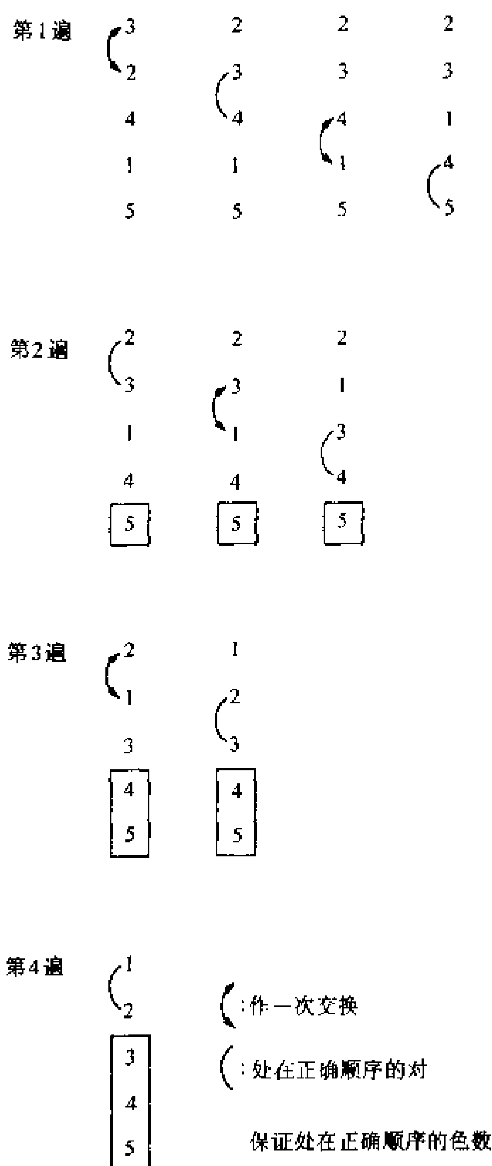


图 8-35　冒泡排序的步骤

在算法 1 里给出冒泡排序的伪代码。

冒泡排序的效率如何？因为在第 i 遍使用 $n-i$ 次比较，所以在 n 个元素的列表的冒泡排序里所使用的总比较次数是

$$(n-1)+(n-2)+\cdots+2+1$$

这是 $n-1$ 个最小整数之和。根据 3.2 节例 9，它等于 $(n-1)n/2$。所以，冒泡排序使用 $n(n-1)/2$ 次比较来排序 n 个元素的列表。（注意冒泡排序总是使用这么多次的比较，因为

即使在某个中间步骤上这个列表变成完全排好顺序的，这个算法仍然继续进行。）因此，冒泡排序算法有最坏情形复杂性 $O(n^2)$。因为对每个正实数 c 来说，对某个充分大的正整数 n 来说有 $n(n-1)/2 > cn \log n$，所以冒泡排序没有 $O(n \log n)$ 的最坏情形时间复杂性。为了达到对最坏情形复杂性的这个最优估计，需要找出另外一个算法。

算法 1　冒泡排序

```
procedure bubblesort(a_1, …, a_n)
for i := 1 to n - 1
begin
    for j := 1 to n - i
        if a_j > a_{j+1} then 交换 a_j 与 a_{j+1}
end
{a_1, …, a_n 为升序}
```

8.4.4　归并排序

许多不同的排序算法都达到了排序算法的最好可能的最坏情形复杂性，即用 $O(n\log n)$ 次比较来排序 n 个元素。在这里将描述这些算法中称为归并排序算法的一个算法。在一般地描述归并排序算法之前，将用一个例子来说明它是如何工作的。

例 3　将用归并排序来排序列表 8，2，4，6，9，7，10，1，5，3。归并排序首先通过不断地一分为二把表分成单个的元素。这个例子的子表的序列表示为图 8-36 上方所示的高度为 4 的平衡二叉树。

排序是通过不断地合并成对的表来完成的。在第一阶段里，把成对的单个元素合并成按升序排列的长度为二的表。然后对成对的表继续进行合并，直到整个表都排成了升序为止。把这些合并成按升序排列的表的序列表示为图 8-36 下方所示的高度为 4 的平衡二叉树（注意这个树是“上下颠倒”显示的）。■

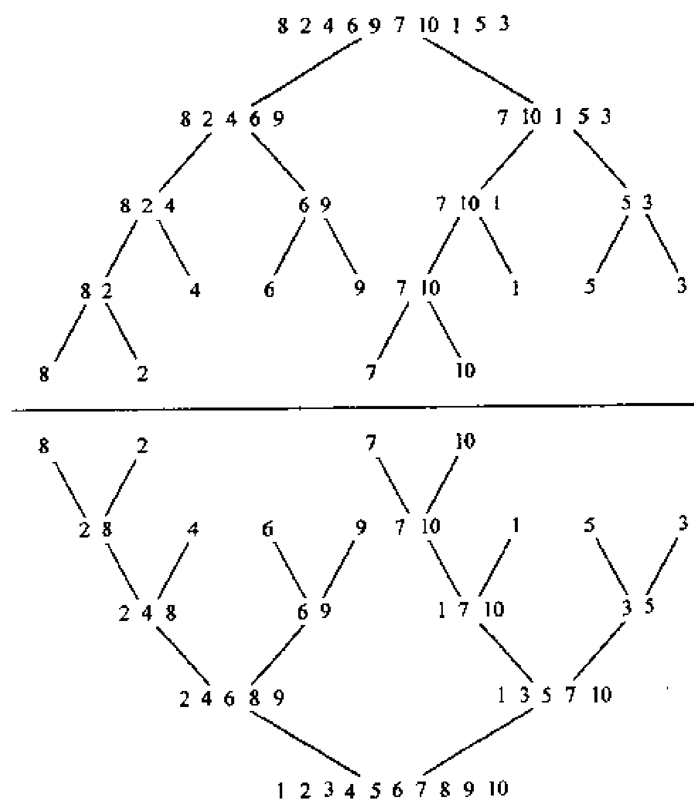


图 8-36　对 8，2，4，6，9，7，10，1，5，3 的归并排序

在一般情况下，归并排序是这样进行的：反复地把表分成长度相等的两个子表（或者其中一个子表比另一个子表多一个元素），直到每个子表包含一个元素为止。这

些子表的序列可以表示成平衡的二叉树。这个过程这样继续进行：不断地合并成对的子表，其中的两个列表都是按升序排列的，把它们合并成元素都是按升序排列的较大的列表，直到原来的列表排成升序为止。这些合并的子表的序列可以表示成平衡的二叉树。

也可以递归地描述归并排序。为了作归并排序，把表分成大小相等或近似相等的两个子表，用归并排序算法排序每个子表，然后合并这两个子表。把给出归并排序递归形式的完整说明留给读者。

为了实现归并排序，需要把两个有序列表合并成更大有序列表的有效算法。现在将描述这样的过程。

例 4 描述如何合并两个列表 2，3，5，6 和 1，4。表 7-1 说明所使用的步骤。

表 7-1 合并已排序的列表 2，3，5，6 和 1，4

第一个列表	第二个列表	合并的列表	比较
2 3 5 6	1 4		1<2
2 3 5 6	4	1	2<4
3 5 6	4	1 2	3<4
5 6	4	1 2 3	4<5
5 6		1 2 3 4	
		1 2 3 4 5 6	

首先，比较两个列表里的最小元素，它们分别是 2 和 1。因为 1 较小，所以把它放在合并的列表的开头，并且从第二个表删除它。在这个阶段，第一个表是 2，3，5，6，第二个表是 4，组合而成的表是 1。

其次，比较 2 和 4，它们是两个表的最小元素。因为 2 较小，所以添加它到组合的表，并且从第一个表删除它。在这个阶段，第一个表是 3，5，6，第二个表是 4，组合而成的表是 1，2。

继续比较 3 和 4，它们是各自表里的最小元素。因为 3 是这两个元素中较小的，所以添加它到组合的表，并且从第一个表删除它。在这个阶段，第一个表是 5，6，第二个表是 4，组合而成的表是 1，2，3。

然后比较 5 和 4，它们是两个表里的最小元素。因为 4 这两个元素中较小的，所以添加它到组合的表，并且从第二个表删除它。在这个阶段，第一个表是 5，6，第二个表是空的，组合而成的表是 1，2，3，4。

最后，因为第二个表是空的，所以第一个表的所有元素可以附加到组合表的后面，保持它们在第一个表里的出现顺序。这样就产生出有序表 1，2，3，4，5，6。 ■

现在考虑合并两个有序列表 L_1 和 L_2 成一个有序列表 L 的一般问题。可以使用下面的过程。从空表 L 开始。比较两个表的最小元素。把这两个元素中较小的放到 L 后面，并且从它所在的表删除它。下一步，若 L_1 和 L_2 有一个是空的，则附加另一个（非空）表到 L，这样就完成了合并。若 L_1 和 L_2 都非空，则重复这个过程。算法 2 给出这个过程的伪代码描述。

在对归并排序的分析中，将需要估计合并两个有序列表 L_1 和 L_2 所用的比较次数。对于算法 2 来说，容易得出这样的估计。每次比较 L_1 的一个元素与 L_2 的一个元素，把一个附加元素添加到合并的列表 L 中。不过，当 L_1 或 L_2 为空时，就不需要更多的比较了。因

此，当执行 $m+n-2$ 次比较（其中 m 和 n 分别是 L_1 和 L_2 中的元素个数）时，算法 2 处于最低的效率，在 L_1 和 L_2 的每个当中只剩下一个元素。下一次比较将是所需要的最后一次，因为这次比较使得这两个表之一为空。因此，算法 2 使用不超过 $m+n-1$ 次比较。下面的引理总结了这个估计。

算法 2 合并两个列表

```
procedure merge(L1,L2:列表)
L :=空表
while  L1 和 L2 都不是空的
begin
  从 L1 和 L2 的第一个元素中删除较小的一个并且把它放到 L 后面
  if 删除这个元素导致一个表为空 then 从另外一个表删除所有元素
       并且把它们附加到 L
end
{L 是元素按升序排列的合并的列表}
```

引理 1 使用不超过 $m+n-1$ 次比较，可以把带 m 个元素和 n 个元素的两个排序的列表合并成一个排序的列表。

有时使用远远少于 $m+n-1$ 次比较就可以合并两个长度为 m 和 n 的排序的列表。例如，当 $m=1$ 时，可以用二叉搜索过程来把第一个表里的这一个元素放进第二个表。这只需要 $\lceil \log n \rceil$ 次比较，对 $m=1$ 来说，$\lceil \log n \rceil$比 $m+n-1=n$ 小得多。在另一方面，对 m 和 n 的某些值来说，引理 1 给出了最好可能的界限。即存在着带有 m 个和 n 个元素的表，不能用少于 $m+n-1$ 次比较来合并它们。（见本节末尾的练习 7。）

现在可以分析归并排序的复杂性了。代替研究一般问题的是，将假定表中的元素个数 n 是 2 的幂，比方说 2^m。这样将使得分析不是太复杂，但是当实际情况不是这样时，还可以做各种修改，这些修改将产生同样的估计。

在分解过程的第一阶段，把表分解成两个子表，每个子表都有 2^{m-1}个元素，位于分解所生成的树的 1 层上。这个过程继续下去，把两个带 2^{m-1}个元素的子表分解成四个在 2 层上各有 2^{m-2}个元素的子表，并依次类推。在一般情况下，在 $k-1$ 层上有 2^{k-1}个表，每个表有 2^{m-k+1}个元素。在 $k-1$ 层上的这些表分解成在 k 层上的 2^k 个表，每个表有 2^{m-k}个元素。在这个过程的最后，有 2^m 个表，每个表在 m 层有一个元素。

这样来开始合并：把 2^m 个有一个元素的表成对地组合成 2^{m-1}个表，都在 $m-1$ 层上，各有两个元素。为了这样做，把 2^{m-1}对有一个元素的表合并。每一对表的合并需要恰好两次比较。这个过程继续下去，使得在 k 层上（$k=m$，$m-1$，$m-2$，…，3，2，1），2^k 个各有 2^{m-k}个元素的表合并成 2^{k-1}个表，各有 2^{m-k+1}个元素，都在 $k-1$ 层上。为了这样做，需要总共 2^{k-1}次合并两个表，每个表有 2^{m-k}个元素。但是，根据引理 1，这些合并每个都可以用至多 $2^{m-k}+2^{m-k}-1=2^{m-k+1}-1$ 次比较来完成。因此，从 k 层进行到 $k-1$ 层，可以用至多 2^{k-1}（$2^{m-k+1}-1$）次比较来完成。对所有这些估计求和就证明了归并排序所需要的比较次数至多是

$$\sum_{k=1}^{m} 2^{k-1}(2^{m-k+1}-1) = \sum_{k=1}^{m} 2^m - \sum_{k=1}^{m} 2^{k-1} = m2^m - (2^m - 1) = n\log n - n + 1$$

因为 $m=\log n$ 和 $n=2^m$。(这样求 $\sum_{k=1}^{m}2^m$ 的值：注意它是 m 个相同的项之和，每个都等于 2^m。这样求 $\sum_{k=1}^{m}2^{k-1}$ 的值：用 3.2 节例 6 几何级数各项求和的公式。)

这样的分析说明，归并排序达到了排序算法所需比较次数的最好可能的大 O 估计，如下面的定理所述。

定理 2　对一个带 n 个元素的列表进行归并排序所需要的比较次数是 $O(n\log n)$。

在练习中描述另一个有效的排序算法——快速排序。

练习

1. 用冒泡排序来排序 3，1，5，7，4，说明在每步上所获得的列表。
2. 用冒泡排序来排序 d，f，k，m，a，b，说明在每步上所获得的列表。
*3. 修改冒泡排序算法，使得当不再需要交换时，算法就停止。用伪代码来表达这个更有效的算法形式。
4. 用归并排序来排序 4，3，2，5，1，8，7，6。说明算法所用的所有步骤。
5. 用归并排序来排序 b，d，a，f，g，h，z，p，o，k。说明算法所用的所有步骤。
6. 为了用算法 2 来合并下面的成对的列表，需要多少次比较？
   a) 1，3，5，7，9；2，4，6，8，10
   b) 1，2，3，4，5；6，7，8，9，10
   c) 1，5，6，7，8；2，3，4，9，10
7. 证明：存在着带有 m 个和 n 个元素的列表，使得它们不能用算法 2 以少于 $m+n-1$ 次的比较来合并成一个排序的列表。
*8. 当两个升序的列表里的元素个数如下时，把它们合并成一个升序的表，所需要的最少比较次数是什么？
   a) 1，4　　b) 2，4　　c) 3，4　　d) 4，4

选择排序首先找出表中的最小元素。把这个元素移到前面。然后找出剩余元素里的最小元素并且把它放到第二个位置。重复这个过程，直到整个表都已经排序了为止。

9. 用选择排序来排序下面的列表。
   a) 3，5，4，1，2　　b) 5，4，3，2，1　　c) 1，2，3，4，5
10. 用伪代码写出选择排序算法。
11. 执行 n 个元素的选择排序需要多少次比较？

快整排序是一个有效算法。为了排序 a_1，a_2，…，a_n，这个算法首先挑出第一个元素 a_1 并且构造两个子表，第一个子表包含小于 a_1 的元素，按照元素出现的顺序排列。第二个子表包含大于 a_1 的元素，按照元素出现的顺序排列。然后把 a_1 放在第一个子表的后面。对每个子表递归地重复这个过程，直到所有子表都只包含一个项为止。n 个项的有序表是这样获得的：按照只含有一个项的子表出现的顺序来组合它们。

12. 用快速排序来排序 3，5，7，8，1，9，2，4，6。
13. 设 a_1，a_2，…，a_n 是 n 个不同实数的列表。从这个表构造两个子表，第一个子表包含小于 a_1 的元素而第二个子表包含大于 a_1 的元素，那么需要多少次比较？

14. 用伪代码描述快速排序算法。
15. 用快速排序算法来排序 4 个元素的表，需要的最大比较次数是什么？
16. 用快速排序算法来排序 4 个元素的表，需要的最小比较次数是什么？
17. 就所用的比较次数而言，确定快速排序算法的最坏情形复杂性。
*18. 证明：对 $n>4$ 来说，$\log n!$ 大于 $(n\log n)/4$。
　[提示：从不等式 $n!>n(n-1)(n-2)\cdots\lceil n/2\rceil$ 入手。]
*19. 用伪代码写出归并排序算法。

8.5　生成树

8.5.1　引言

考虑一下图 8-37 a）所示的简单图所表示的缅因州的道路系统。在冬天里保持这些道路通畅的唯一方式就是经常扫雪。高速公路部门希望对最少的道路扫雪，使得总是存在连接任何两个城镇的干净道路总是干净的。如何才能做到这一点？

至少必须对 5 条道路扫雪才能保证在任何两个城镇之间有一条道路。图 8-37 b）说明了这样一些道路。注意表示这些道路的子图是一棵树，因为它是连通的并且包含 6 个顶点和 5 条边。

这个问题是用包含原来简单图的所有顶点的边数最少的连通子图来解决的。这样的图必然是树。

定义 1　设 G 是简单图。G 的生成树是包含 G 的每个顶点的 G 的子图。

有生成树的简单图必然是连通的，因为在任何两个顶点之间都有生成树里的通路。反过来也是对的；即每个连通图都有生成树。在证明这个结果之前将给出一个例子。

例 1　找出图 8-37 所示简单图的生成树。

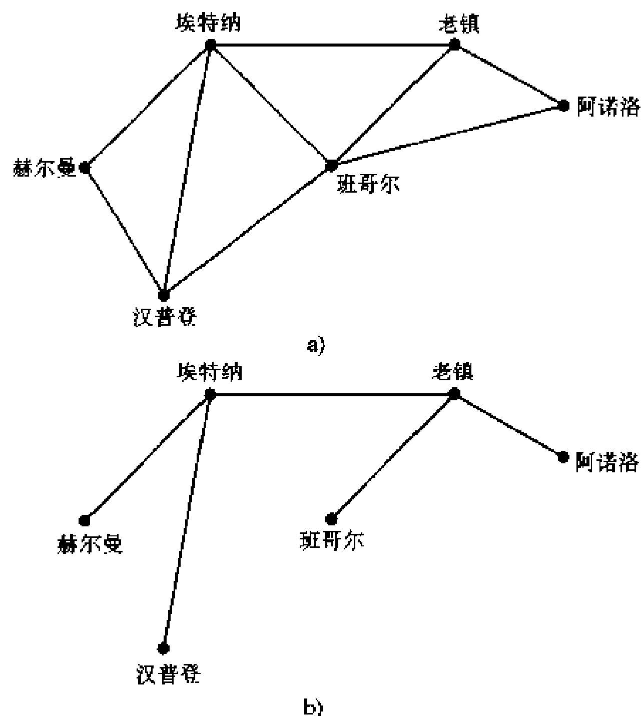


图 8-37　a）道路系统和　b）需要扫雪的一些道路

解　图 G 是连通的，但是它不是树，因为它包含简单回路。删除边 $\{a, e\}$。这样就消除了一个简单回路，而且所得出的子图仍然是连通的并且仍然包含 G 的每个顶点。其次删除边 $\{e, f\}$ 以便消除第二个简单回路。最后，删除边 $\{c, g\}$ 以便产生一个没有简单回路的简单图。这个子图是生成树，因为它是包含 G 的每个顶点的树。在图 8-39 中说明用来产生这个生成树的边的删除序列。

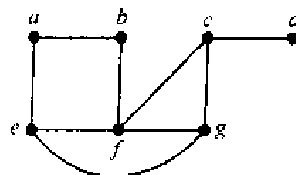


图 8-38　简单图 G

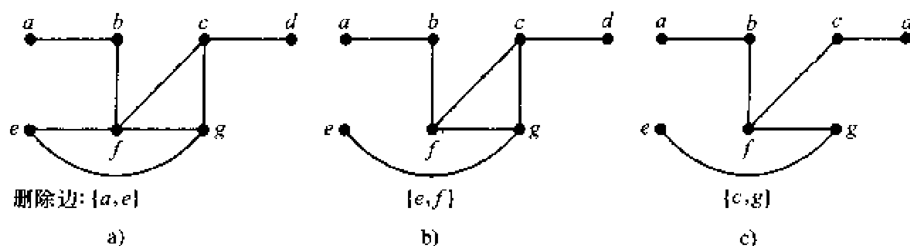


图 8-39　通过删除边形成简单回路来产生 G 的一个生成树

在图 8-39 中所示的生成树不是唯一的 G 的生成树。例如，图 8-40 所示的每个树都是 G 的生成树。■

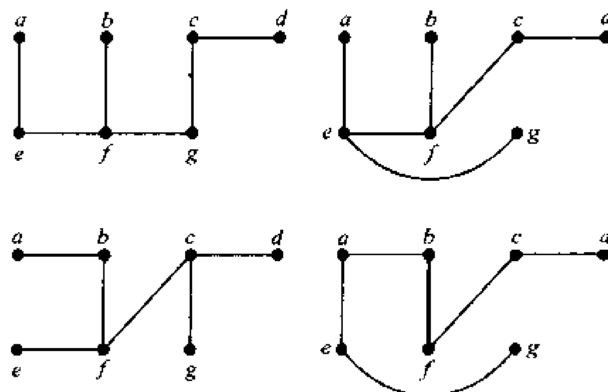


图 8-40　G 的一些生成树

定理 1　简单图是连通的当且仅当它具有生成树。

证　首先，假定简单图 G 有生成树 T。T 包含 G 的每个顶点。另外，在 T 的任何两个顶点之间都有 T 里的通路。因为 T 是 G 的子图，所以在 G 的任何两个顶点之间都有 T 里的通路。因此，G 是连通的。

现在假定 G 是连通的。若 G 不是树，则它必然包含简单回路。从这些简单回路中的一个里删除一条边。所得出的子图少了一条边，但是仍然包含 G 的所有顶点并且是连通的。若这个子图不是树，则它有简单回路；所以像前面那样，删除一个简单回路里的一条边。重复这个过程直到没有简单回路剩下为止。这是可能的，因为在图里只有有穷的边数。当没有简单回路剩下时，这个过程终止。产生出一棵树，因为在删除边时这个图保持连通。这个树是生成树，因为它包含 G 的每个顶点。■

下面的例子说明，在数据网络里生成树是重要的。

例2 IP组播 在互联网协议（IP）网络上的组播里，生成树起到重要作用。为了从源计算机发送数据到多个接收计算机，每个接收计算机是一个子网，可以分别发送数据到每个计算机。称为单播的这种类型的网络是无效的，因为在网络上发送相同数据的许多副本。为了让传送数据到多个接收计算机更友效，就使用IP组播。在IP组播里，一个计算机在网络上发送数据的单一副本，当数据达到中间路由器时，就把数据分发到一个或更多的其他路由器，以便接收计算机都在它们不同的子网里最终接收到这些数据。(路由器是专门在网络里子网之间分发IP数据报的计算机。在组播里，路由器使用D类地址，每个都表示接收计算机可以参加的一个会话；见第4.3节里例8。)

为了让数据尽可能快地到达接收计算机，在数据穿过网络的通路里就不应当存在环路(在图论中称它们是回路)。即一旦数据已经到过一个具体的路由器，就不应当再返回这个路由器。为了避免环路，组播路由器用网络算法来构造下图的生成树，这个图以组播源路由器和包含接收计算机的子网来作为顶点，以边表示计算机和（或）路由器之间的连接。这个生成树的根就是组播源。包含接收计算机的子网就是这个树的树叶。(注意不包含接收计算机的子网都不包含在这个图里。) 在图8-41中说明这些内容。 ■

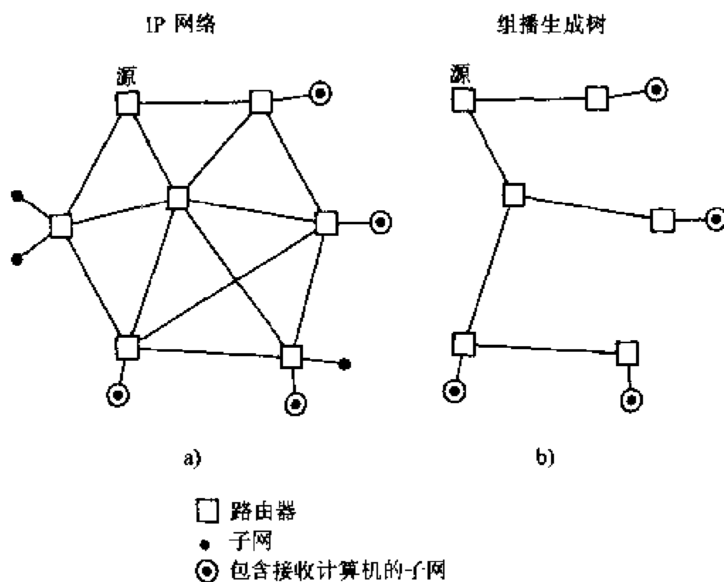


图8-41 一个组播生成树

8.5.2 一些构造生成树的算法

定理1的证明给出通过从简单回路删除边来找出生成树的算法。这个算法是无效的，因为它要求找出简单回路。代替通过删除边来构造生成树，也可通过相继地添加边来建立生成树。在这里将给出基于这个原理的两个算法。

可以用深度优先搜索来建立连通简单图的生成树。将形成一个根树，而生成树将是这个根树的无向底图。任意选择图中一个顶点作为根。通过相继地添加边来形成在这个顶点上开始的通路，其中每条新边都与通路上的最后一个顶点以及还不在通路上的一个顶点相关联。继续尽可能地添加边到这条通路。若这条通路经过图的所有顶点，则由这条通路组成的树就

是生成树。不过，若这条通路没有经过图的所有顶点，则必须添加其他的边。后退到通路里的次最后顶点，若有可能，则形成在这个顶点上开始的经过还没有访问过的顶点的通路。若不能这样做，则后退到通路里的另外一个顶点，即在通路里后退两个顶点，然后再试。重复这个过程，在所访问过的最后一个顶点上开始，在通路上一次后退一个顶点，只要有可能就形成新的通路，直到不能添加更多的边为止。因为这个图有有穷的边数并且是连通的，所以这个过程以产生生成树而告终。在这个算法的一个阶段上是通路末端的顶点将是根树里的树叶，而在其上开始构造一条通路的顶点将是内点。读者应当注意到这个过程的递归本质。另外，注意若图中的顶点是排序的，则当总是选择在该顺序里可用的第一个顶点时，在这个过程的每个阶段上对边的选择就全都是确定的。不过，将不总是明显地对图的顶点排序。

深度优先搜索也称为*回溯*，因为这个算法返回以前访问过的顶点以便添加路径。下面的例子说明回溯。

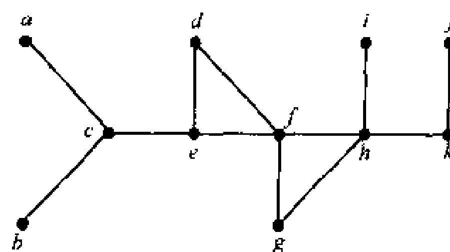


图 8-42　图 G

例 3　用深度优先搜索来找出图 8-42 所示图 G 的生成树。

解　在图 8-43 中显示深度优先搜索为了产生 G 的生成树而使用的步骤。任意地从顶点 f 开始。一条通路是这样建立的：相继地添加与还不在通路上的顶点相关联的边，只要有可能就这样做。这样就产生通路 f，g，h，k，j（注意也可能建立其他的通路）。下一步，回溯到 k。不存在从 k 开始包含还没有访问过的顶点的通路。所以回溯到 h。形成通路 h，i。然后回溯到 h，然后再回溯到 f。从 f 建立通路 f，d，e，c，a。然后再回溯到 c 并且形成通路 c，b。这样就产生了生成树。■

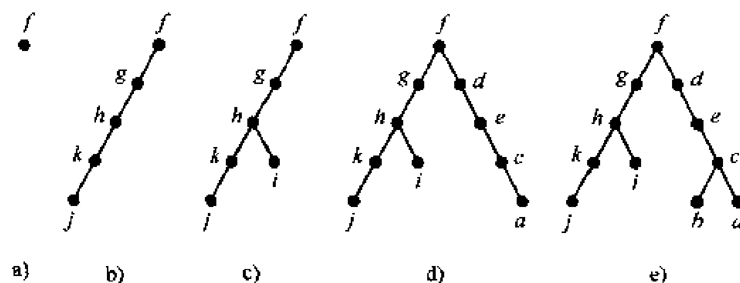


图 8-43　G 的深度优先搜索

也可以通过使用*宽度优先搜索*来产生简单图的生成树。同样，将构造一个根树，而这个根树的无向底图就形成生成树。从图的顶点中任意地选择一个根。然后添加与这个顶点相关联的所有边。在这个阶段所添加的新顶点成为生成树里在 1 层上的顶点。任意地排序它们。下一步，按顺序访问 1 层上的每个顶点，只要不产生简单回路，就添加与这个顶点相关联的每条边到树里。这样就产生了树里在 2 层上的顶点。遵循相同的过程，直到已经添加了树里的所有顶点。这个过程将会终止，因为在图中只有有穷的边数。这就产生了生成树，因为已经产生了包含图中每一个顶点的树。下面是深度优先搜索的一个例子。

例 4　用深度优先搜索来找出图 8-44 所示的图的生成树。

解 在图 8-45 中显示深度优先搜索过程的各步骤。选择顶点 e 作为根。然后添加与 e 相关联的所有边，所以添加了从 e 到 b，d，f 和 i 的边。这四个顶点都是在树的 1 层上。下一步，添加从这些在 1 层上的顶点到还不在树上的相邻顶点的边。因此，添加从 b 到 a 和 c 的边，以及从 d 到 h、从 f 到 j 和 g、从 i 到 k 的边。新顶点 a，c，h，j，g 和 k 都是在 2 层上。下一步，添加从这些顶点到还不在树上的相邻顶点的边。这样就添加从 g 到 l 以及从 k 到 m 的边。■

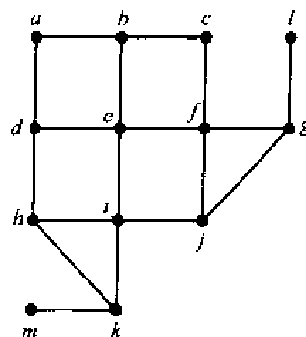


图 8-44 图 G

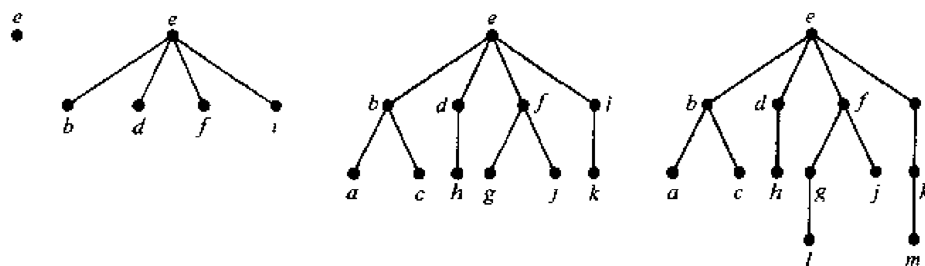


图 8-45 G 的宽度优先搜索

8.5.3 回溯

存在这样的问题，只能通过执行对所有可行解的穷举搜索来解决它。系统地搜索出一个解的一种方式是使用决策树，其中每个内点都表示一次决策。而每个树叶都表示一个可行解。为了通过回溯来求出一个解，首先尽可能地做出一系列决策来尝试得出一个解。可以用决策树里的通路来表示决策序列。一旦知道了决策序列的任何扩展都不能得出解，就回溯到当前顶点的父亲顶点，并且若有可能，就用另外一个决策序列来尝试得出一个解。继续这个过程，直到找到一个解，或者证明没有解存在为止。下面的例子说明回溯的用途。

例 5 **图着色** 如何用回溯来判定是否可以用 n 种颜色给一个图着色？

解 以下面的方式用回溯来解决这个问题。首先选择某个顶点 a 并且指定它的颜色为 1。然后挑选第二个顶点 b，而且若 b 不与 a 相邻，则指定它颜色为 1。否则，指定 b 颜色为 2。然后来到第三个顶点 c。若有可能，则对 c 用颜色 1；否则，若有可能则用颜色 2。只有当颜色 1 和颜色 2 都不能用时才应当用颜色 3。继续这个过程，只要有可能就指定 n 种颜色中的一种给每个新顶点，总是使用表中第一种允许的颜色。若遇到不能用 n 种颜色中任何一种来着色的顶点，则回溯到最后一次所做的指定，并且若有可能就改变最后着色的顶点的颜色，用表中下一种允许的颜色。若不可能改变这个颜色，则再回溯到更前面的指定，一次后退一步，直到有可能改变一个顶点的颜色为止。然后只要有可能就继续指定新顶点的颜色。若使用 n 种颜色的着色存在，则回溯将产生它。（不幸的是这个过程是极其低效的。）

具体地说，考虑用三种颜色来着色图 8-46 所示的图的问题。图 8-46 所示的树说明如何用回溯来构造 3 着色。在这个过程里，首先用红色，其次用蓝色，最后用绿色。显然可以不用回溯来解决这个简单的例子，但是它是对这个技术的一个好的说明。

在这个树里，从根开始表示指定红色给 a 的最初的通路，导致 a 红色、b 蓝色、c 红色

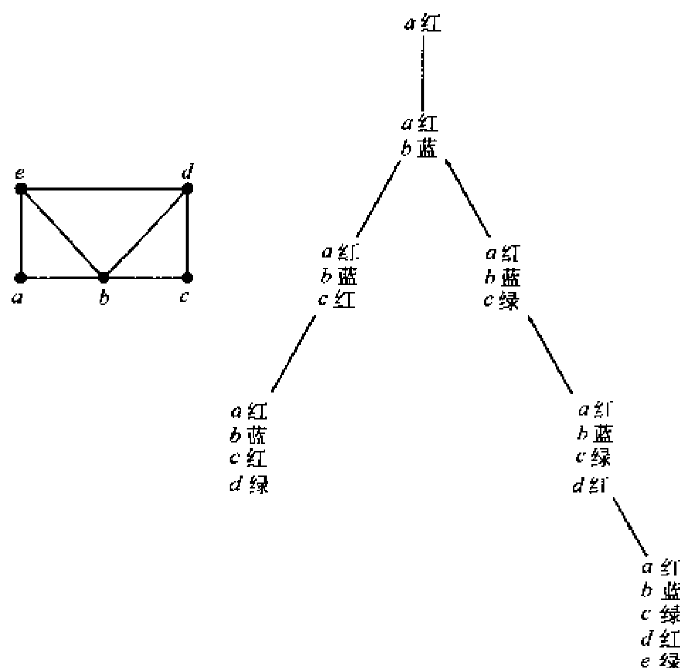


图 8-46　用回溯来给一个图着色

而 d 绿色的着色。当以这种方式来着色 a，b，c 和 d 时，就不可能用这三种颜色中的任何一种来着色 e。所以，回溯到表示这个着色的顶点的父亲。因为没有其他颜色可以用在 d 上，所以再回溯一层。然后改变 c 的颜色为绿色。通过接着指定红色给 d 和绿色给 e，就获得这个图的着色。■

例 6　**n 皇后问题**　n 皇后问题问：在 $n \times n$ 棋盘上如何放置 n 个皇后，使得没有两个皇后可以互相攻击。如何用回溯来解决 n 皇后问题？

解　为了解决这个问题，必须在 $n \times n$ 棋盘上找出 n 个位置，使得这些位置中没有两个是在同一行上、同一列上或在同一斜线上（斜线是由对某个 m 来说满足 $i+j=m$ 或对某个 m 来说满足 $i-j=m$ 的所有位置 (i, j) 组成的）。将用回溯来解决 n 皇后问题。从空棋盘开始。在 $k+1$ 阶段上，尝试在棋盘上第 $k+1$ 列里放置一个新皇后，其中在前 k 列里已经有了皇后。检查第 $k+1$ 列里的格子，从第一行的格子开始，寻找放置这个皇后的位置，使得它不与已经在棋盘上的皇后在同一行里或在同一斜线上。（已经知道它不在同一列里。）若不可能在第 $k+1$ 列里找到放置皇后的位置，则回溯到在第 k 列里对皇后的放置。在这一列里下一个允许的行里放置皇后，若这样的行存在的话。若没有这样的行存在，则继续回溯。

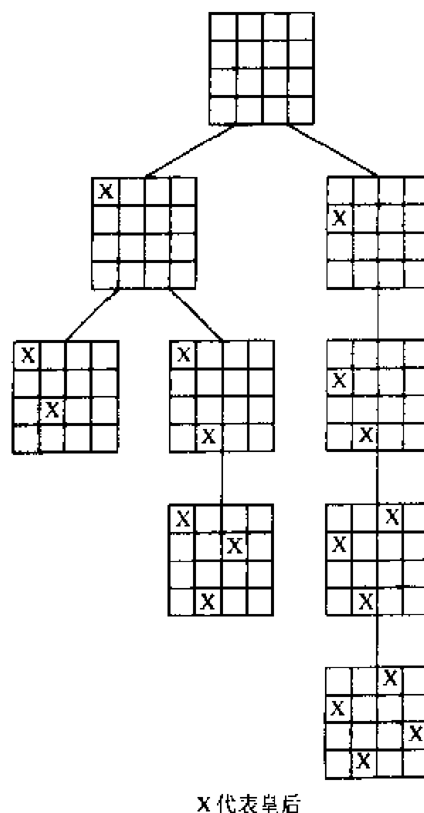


图 8-47　四皇后问题的回溯解法

具体地说，图 8-47 显示四皇后问题的回溯解法。在这种解法里，在第一行第一列里放置一个皇后。然后在第二列的第三行里放置一个皇后。不过，这样就使得不可能在第三列里

放置一个皇后。所以就回溯并且在第二列的第四行里放置一个皇后。当这样做时，就可以在第三列的第二行里放置一个皇后。但是没有办法在第四列里添加一个皇后。这说明当在第一行第一列里放置一个皇后时就得不出解。回溯到空棋盘，在第一列的第二行里放置一个皇后。这样就得出图 8-47 所示的解。■

例 7　子集和　考虑下面的问题。给定一组正整数 x_1，x_2，…，x_n，求这组整数的和为 M 的一个子集。如何用回溯来解决这个问题？

解　从空无一项的和来开始。通过相继地添加项来构造这个和。若当添加这个序列里的一个整数到和里、而这个和仍然小于 M 时，则包含这个整数。若得出使得添加任何一项就大于 M 的一个和，则通过去掉这个和的最后一项来回溯。

图 8-48 显示如下问题的回溯解法：求{31,27,15,11,7,5}的和等于 39 的子集。■

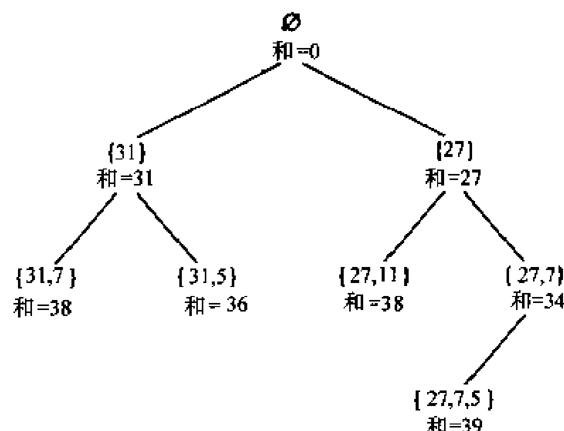
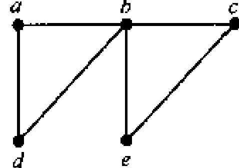


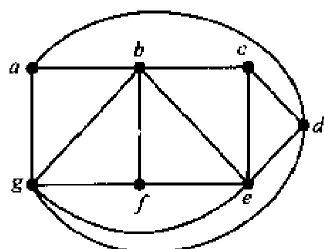
图 8-48　用回溯求等于 39 的和

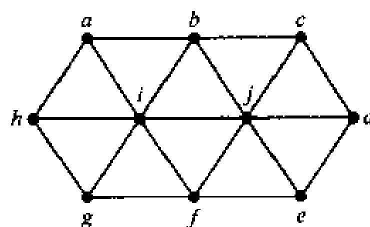
练习

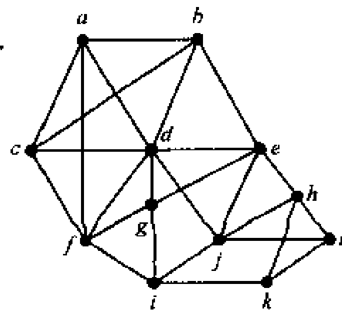
1. 为了产生生成树，必须从带有 n 个顶点和 m 条边的连通图里删除多少条边？

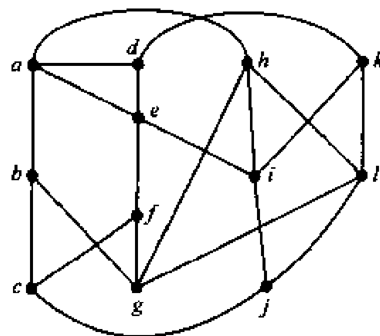
在练习 2～6 中，通过删除简单回路里的边来求所示的图的生成树。

2.


3.


4.


5.


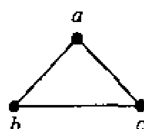
6.


7. 求下面每个图的生成树。

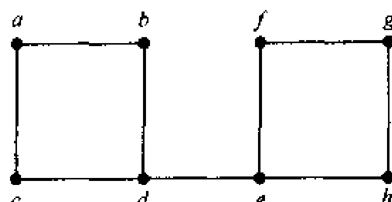
a) K_5 b) $K_{4,4}$ c) $K_{1,6}$ d) Q_3 e) C_5 f) W_5

在练习 8～10 中，画出所给的简单图的生成树。

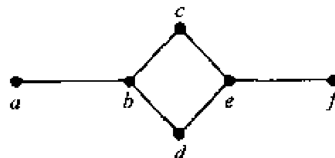
8.



9.



10.



*11. 下面的每个简单图各有多少个不同的生成树？

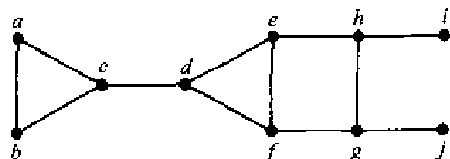
a) K_3 b) K_4 c) $K_{2,2}$ d) C_5

*12. 下面的每个简单图各有多少种不同构的生成树？

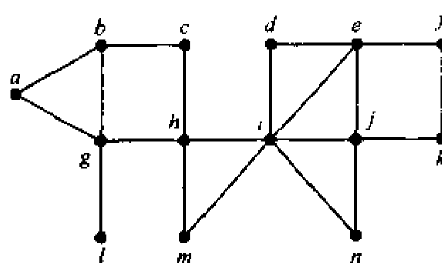
a) K_3 b) K_4 c) K_5

在练习 13～15 中，用深度优先搜索来构造所给的简单图的生成树。选择 a 作为这个生成树的根并且假定顶点都以字母顺序来排序。

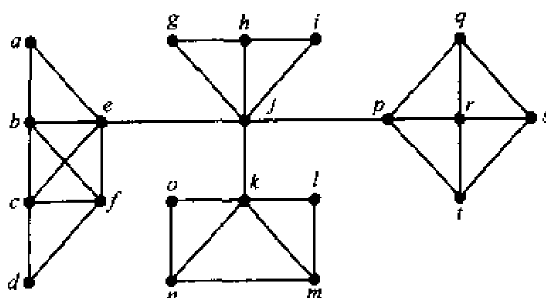
13.



14.



15.



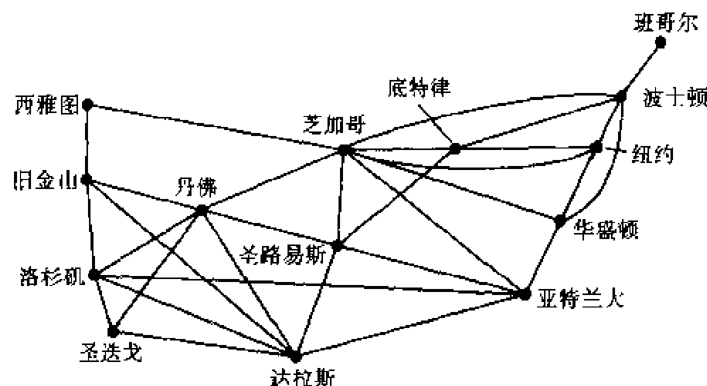
16. 用宽度优先搜索来构造练习 13～15 中每个简单图的生成树。选择 a 作为每个生成树的根。

17. 假定一家航空公司必须压缩它的飞行时间表以节省资金。若它原来的航线如下页图所示，则可以中断哪些航班以保持在所有城市对之间的服务（其中从一个城市飞往另外一个城市可能需要换乘飞机）？

18. 何时连通简单图里的一条边必然在这个图的每个生成树里？

19. 哪些连通简单图恰好有一个生成树？

20. 解释一下如何用宽度优先搜索或深度优先搜索来排序连通图的顶点。



*21. 用伪代码写出深度优先搜索过程。

*22. 用伪代码写出宽度优先搜索过程。

*23. 证明：在连通简单图里顶点 v 和 u 之间的最短通路的长度，等于在以 v 为根的 G 的宽度优先生成树里 u 的层数。

24. 用回溯来试验找出使用三种颜色对 7.8 节练习 5～7 中每个图的着色。

25. 用回溯来对下面的 n 值解决 n 皇后问题。

 a) $n=3$ b) $n=5$ c) $n=6$

26. 用回溯来求集合 {27，24，19，14，11，8} 的和为下列值的子集，若存在的话。

 a) 20 b) 41 c) 60

27. 解释一下如何用回溯来找出图中的哈密顿通路或哈密顿回路。

28. a) 解释一下如何用回溯来找出迷宫的出路，给定出发位置和出口位置。考虑把迷宫划分成位置，其中在每个位置上可行的移动包括四种可能性（上，下，右，左）之一。

 b) 找出在下面的迷宫里从标记 X 的出发位置到出口的通路。

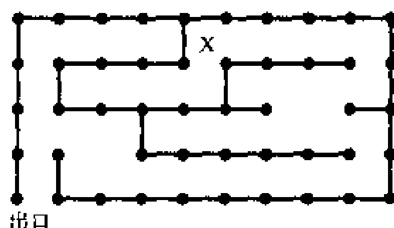


图 G 的生成森林是包含 G 的每个顶点的森林，使得当在两个顶点之间有 G 里的通路时，这两个顶点就在同一个树里。

29. 证明：每个有穷简单图都有生成森林。

30. 在图的生成森林里有多少棵树?

31. 对带有 n 个顶点 m 条边和 c 个连通分支的图来说，必须删除多少条边才能产生它的生成森林?

32. 设计基于删除形成简单回路的边来构造图的生成森林的算法。

33. 设计基于深度优先搜索来构造图的生成森林的算法。

34. 设计基于宽度优先搜索来构造图的生成森林的算法。

设 T_1 和 T_2 都是一个图的生成树。在 T_1 和 T_2 之间的距离是在 T_1 和 T_2 里但不是 T_1 和 T_2 所共有的边的数目。

35. 求图 8-38 所示的图 G 在图 8-39 c) 和图 8-40 中所示的每对生成树之间的距离。

*36. 假定 T_1，T_2 和 T_3 都是简单图 G 的生成树。证明：在 T_1 和 T_3 之间的距离不超过 T_1 和 T_2 之间的距离与 T_2 和 T_3 之间的距离的和。

*37. 假定 T_1 和 T_2 都是简单图 G 的生成树。另外，假定 e_1 是在 T_1 但不在 T_2 里的一条边。证明：存在着在 T_2 但不在 T_1 里的一条边 e_2，使得若从 T_1 删除 e_1 而添加 e_2 到 T_1 里，则 T_1 仍然是生成树，并且若从 T_2 删除 e_2 而添加 e_1 到 T_2 里，则 T_2 仍然是生成树。

*38. 证明：通过相继地删除一条边而添加另外一条边，就有可能从任何一个生成树得出一个生成树的序列。

有向图的*根生成树*是由这个图的边组成的根树，使得这个图的每个顶点都是树里一条边的终点。

39. 对 7.5 节练习 24～28 的每个有向图来说，求这个图的根生成树，或者确定不存在这样的树。

*40. 证明：每个顶点的入度和出度都相等的连通有向图有根生成树。[提示：使用欧拉回路。]

*41. 给出构造每个顶点的入度和出度都相等的连通有向图的根生成树的算法。

8.6　最小生成树

8.6.1　引言

一家公司计划建立连接它的五个计算机中心的通信网络。可以用租用的电话线连接这些中心的任何一对。应当建立哪些连接，以便保证在任何两个计算机中心之间都有通路，使得网络的总成本是最小的？可以用图 8-49 所示的带权图为这个问题建模，其中顶点

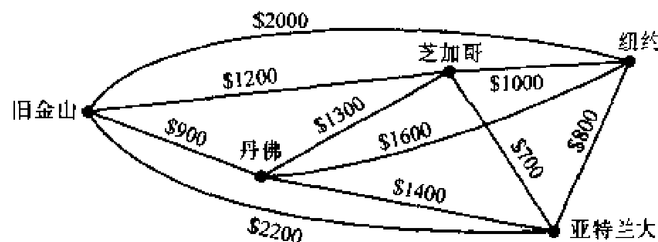


图 8-49　说明一个计算机网络的线路月租费的带权图

表示计算机中心，边表示可能租用的电话线，边上的权是边所表示的电话线的月租费。通过找出一棵生成树，使得这棵树的各边的权之和为最小，就可以解决这个问题。这样的生成树称为*最小生成树*。

8.6.2　最小生成树算法

有大量的问题可以这样解决：求带权图里一棵生成树，使得这棵树的各边的权之和为最小。

定义 1 连通带权图里的最小生成树是具有最小可能的边的权之和的生成树。

将给出构造最小生成树的两个算法。这两个算法都是通过添加还没有使用过的具有规定性质和权最小的边来进行的。这些算法都是贪心算法的例子。贪心算法是在每个步骤上都做最优选择的算法。在算法的每个步骤上都最优化，并不保证产生全局最优解。不过，本节里给出的构造最小生成树的这两个算法都是产生最优解的贪心算法。

将要讨论的第一个算法是罗伯特·普林[1]在 1957 年给出的，虽然这个算法的基本想法有更早的起源。为了执行普林算法，首先选择带最小权的边，把它放进生成树里。相继向树里添加带最小权的边，这些边与已在树里的顶点相关联，并且不与已在树里的边形成简单回路。当已经添加了 $n-1$ 条边时就停止。

在本节稍后将证明这个算法产生任何连通带权图的最小生成树。算法 1 给出普林算法的伪代码描述。

算法 1 普林算法

```
procedure Prim(G:带 n 个顶点的连通无向图)
T :=权最小的边
for i := 1 to n - 2
begin
    e :=与 T 里顶点相关联的权最小的边，并且若添加到 T 里则不形成简单回路
    T :=添加 e 之后的 T
end {T 是 G 的最小生成树}
```

注意，当有超过一条满足相应条件的带相同权的边时，在算法的这个阶段里对所添加的边的选择就是不确定的。需要排序这些边以便让选择是确定的。在本节剩下的部分将不再考虑这个问题。另外注意，所给的连通带权简单图可能有多于一个的最小生成树。（见练习 9。）下面的例子说明如何使用普林算法。

例 1 用普林算法设计一个具有最低成本的通信网络，这个网络连接图 8-49 中的图所表示的所有计算机。

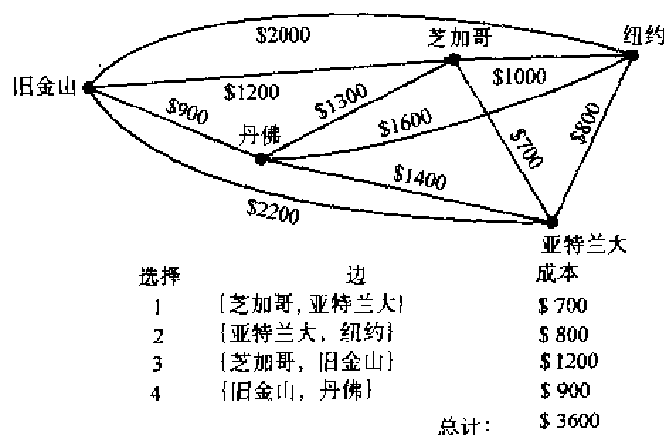
解 这样解决这个问题：求图 8-49 中的图的最小生成树。普林算法是这样执行的：选择权最小的初始边，并且相继添加与树中顶点关联的不形成回路的权最小的边。在图 8-50 中带颜色的边表示普林算法所产生的最小生成树，并且显示出在每个步骤上所做的选择。 ■

例 2 用普林算法求图 8-51 所示的图的最小生成树。

解 用普林算法所构造的最小生成树显示在图 8-52 里。相继选择的边都有显示。 ■

---

[1] 罗伯特·克雷·普林（Robert Clay Prim，生于 1921 年） 罗伯特·普林出生在德克萨斯州的甜水镇，在 1941 年获得电气工程学士，在 1949 年从普林斯顿大学获得数学博士学位。他从 1941 年到 1944 年是通用电气公司的工程师，从 1944 年到 1949 年是美国海军军械实验室的工程师和数学家，从 1948 年到 1949 年是普林斯顿大学的副研究员。他担任过的其他职务有：从 1958 年到 1961 年贝尔电话实验室的数学与力学研究部主任，以及圣地亚公司的研究副总裁。他目前已经退休。



选择	边	成本
1	{芝加哥，亚特兰大}	$ 700
2	{亚特兰大，纽约}	$ 800
3	{芝加哥，旧金山}	$ 1200
4	{旧金山，丹佛}	$ 900
	总计：	$ 3600

图 8-50　图 8-49 里带权图的最小生成树

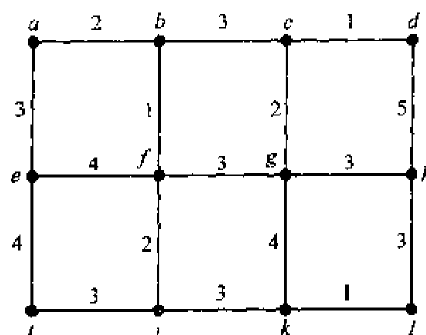
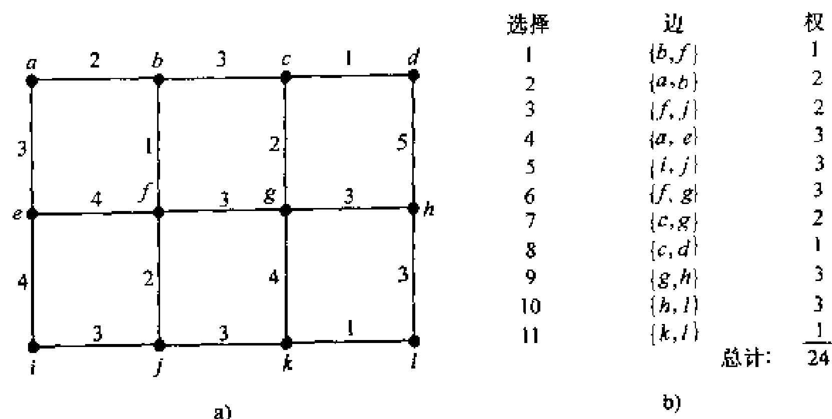


图 8-51　一个带权图

将要讨论的第二个算法是约瑟夫·克鲁斯卡尔[○]在 1956 年发现的，虽然在很早之前就有人描述过它所使用的基本想法。为了执行克鲁斯卡尔算法，要选择图中权最小的一条边[⊖]。



选择	边	权
1	$\{b,f\}$	1
2	$\{a,b\}$	2
3	$\{f,j\}$	2
4	$\{a,e\}$	3
5	$\{i,j\}$	3
6	$\{f,g\}$	3
7	$\{c,g\}$	2
8	$\{c,d\}$	1
9	$\{g,h\}$	3
10	$\{h,l\}$	3
11	$\{k,l\}$	1
	总计：	24

图 8-52　用普林算法构造的最小生成树

相继添加不与已经选择的边形成简单回路的权最小的边。在已经挑选了 $n-1$ 条边之后就停止。

在本节末尾，把证明克鲁斯卡尔算法对每个连通带权图都产生最小生成树留作练习。在算法 2 里给出克鲁斯卡尔算法的伪代码。

读者应当注意普林算法与克鲁斯卡尔算法的区别。在普林算法里，选择与已在树里的一个顶点相关联并且不形成回路的权最小的边；相反，在克鲁斯卡尔算法里，不必选择与已在

○　约瑟夫·伯纳德·克鲁斯卡尔（Joseph Bernard Kruskal 生于 1928 年）　约瑟夫·克鲁斯卡尔出生在纽约城，上了芝加哥大学，并且在 1954 年从普林斯顿大学获得博士学位。他是普林斯顿和威斯康星大学的数学教师，随后是密歇根大学的助理教授。1959 年他成为贝尔实验室的技术委员会成员，他一直担任这个职务。他目前的研究兴趣包括统计语言学和心理测量学。除了他的最小生成树的工作之外，克鲁斯卡尔还因为对多维分级的贡献而著名。当克鲁斯卡尔是二年级的研究生时，他发现了产生最小生成树的算法。他不能肯定他关于这个题目的 $2^1/_2$ 页的论文是否值得发表，但是被其他人说服而递交了它。

⊖　历史注记：约瑟夫·克鲁斯卡尔和罗伯特·普林在 20 世纪 50 年代中期提出他们的算法。不过，他们不是首先发现这些算法的人。例如，人类学家扬·切卡诺夫斯基（Jan Czekanowski）在 1909 年的工作就包含了求最小生成树所需要的许多想法。在 1926 年，奥塔卡·勃鲁乌卡（Otakar Boruvka）在与构造电力网有关的工作中描述了构造最小生成树的方法。

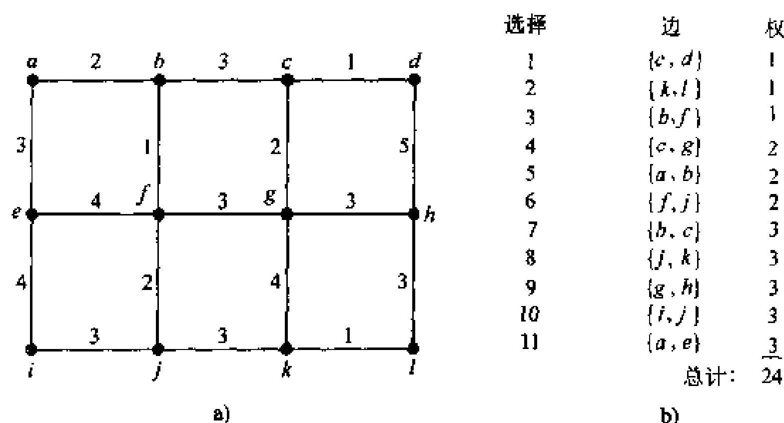
算法 2 克鲁斯卡尔算法

```
procedure Kruskal(G:带 n 个顶点的带权连通无向图)
T :=空图
for i := 1 to n − 1
begin
  e :=当添加到 T 里时不形成简单回路的 G 里权最小的边
  T :=添加 e 之后的 T
end {T 是 G 的最小生成树}
```

树里的一个顶点相关联并且不形成回路的权最小的边。注意，在普林算法里，若没有对边排序，则在这个过程的某个阶段上，对添加的边来说就可能有多于一种的选择。因此，为了让这个过程是确定的，就需要对边进行排序。下面的例子说明如何使用克鲁斯卡尔算法。

例 3 用克鲁斯卡尔算法求图 8-51 所示的带权图里的最小生成树。

解 在图 8-53 里显示这个最小生成树和在克鲁斯卡尔算法每个阶段上对边的选择。■



选择	边	权
1	{c, d}	1
2	{k, l}	1
3	{b, f}	1
4	{c, g}	2
5	{a, b}	2
6	{f, j}	2
7	{b, c}	3
8	{j, k}	3
9	{g, h}	3
10	{i, j}	3
11	{a, e}	3
	总计:	24

图 8-53 克鲁斯卡尔算法产生的最小生成树

现在将证明普林算法产生连通带权图的最小生成树。

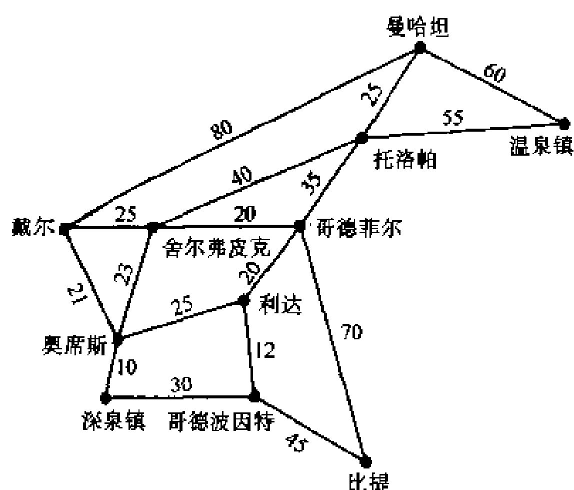
证 设 G 是一个连通带权图。假定普林算法相继地选择的边是 e_1，e_2，…，e_{n-1}。设 S 是以 e_1，e_2，…，e_{n-1} 作为边的树，而设 S_k 是以 e_1，e_2，…，e_k 作为边的树。设 T 是包含边 e_1，e_2，…，e_k 的 G 的最小生成树，其中 k 是满足下列性质的最大整数：存在着包含普林算法所选择的前 k 条边的最小生成树。若证明了 $S=T$，则由此证明了这个定理。

假定 $S\neq T$，所以 $k<n-1$。因此，T 包含边 e_1，e_2，…，e_k，但是不包含 e_{k+1}。考虑由 T 和 e_{k+1} 所组成的图。因为这个图是连通的并且有 n 条边，边太多了就不能是树，所以它必然包含简单回路。这个简单回路必然包含 e_{k+1}，因为在 T 里没有简单回路。另外，在这个简单回路里必然有不属于 S_{k+1} 的边，因为 S_{k+1} 是一棵树。通过从 e_{k+1} 的一个端点开始，该端点也是边 e_1，e_2，…，e_k 之一的端点，并且依循回路直到它到达一条不在 e_{k+1} 里的边为止，就可以找出一条不在 S_{k+1} 里的边 e，它有一个端点也是边 e_1，e_2，…，e_k 之一

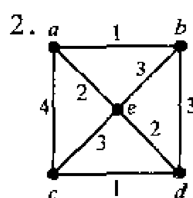
的端点。通过从 T 中删除 e 并且添加 e_{k+1}，就获得带 $n-1$ 条边的树 T'（它是树，因为它没有简单回路）。注意树 T' 包含 e_1，e_2，…，e_k，e_{k+1}。另外，因为普林算法在第 $k+1$ 个步骤上选择 e_{k+1}，并且在这个步骤上 e 也是可用的，所以 e_{k+1} 的权就小于或等于 e 的权。根据这个观察结果就得出 T' 也是最小生成树，因为它的边的权之和不超过 T 的边的权之和。这与对 k 的选择相矛盾，k 是使得包含 e_1，e_2，…，e_k 的最小生成树存在的最大整数。因此，$k=n-1$ 并且 $S=T$。所以普林算法产生最小生成树。 □

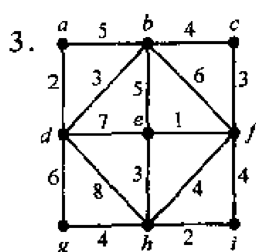
练习

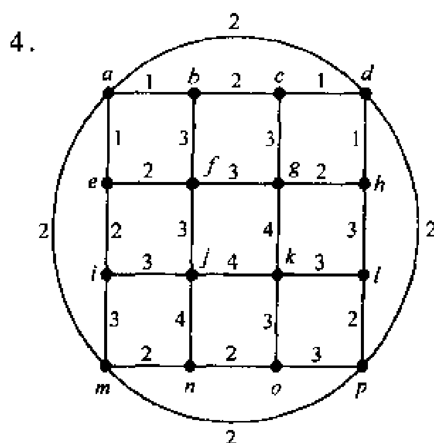
1. 下图所表示的道路都还没有铺设路面。边的权表示在成对的城镇之间的道路长度。哪些道路应当铺设路面，以便在每对城镇之间都有铺设路面的道路，而且使得铺设的道路的长度为最短？（注：这些城镇都在内华达州。）



在练习 2～4 中，用普林算法求所给的带权图的最小生成树。

2. 

3. 

4. 

5. 用克鲁斯卡尔算法设计在本节开头所描述的通信网络。
6. 用克鲁斯卡尔算法求练习 2 中带权图的最小生成树。
7. 用克鲁斯卡尔算法求练习 3 中带权图的最小生成树。

8. 用克鲁斯卡尔算法求练习 4 中带权图的最小生成树。
9. 找出具有多于一棵最小生成树的带有最少可能边数的连通带权简单图。
10. 带权图中的**最小生成森林**是权最小的生成森林。解释如何修改普林算法和克鲁斯卡尔算法来构造最小生成森林。

连通带权无向图的**最大生成树**是带最大可能的权的生成树。

11. 设计与普林算法类似的构造连通带权图的最大生成树的算法。
12. 设计与克鲁斯卡尔算法类似的构造连通带权图的最大生成树的算法。
13. 求练习 2 中带权图的最大生成树。
14. 求练习 3 中带权图的最大生成树。
15. 求练习 4 中带权图的最大生成树。
16. 求在本节开头所提出问题中连接五个计算机中心的次最便宜的通信网络。
*17. 设计求连通带权图中次最短生成树的算法。
*18. 证明：连通带权图中权最小的边，必然属于任何一个最小生成树。
19. 证明：若所有边的权都不相同，则连通带权图中有唯一的最小生成树。
20. 假定连接图 8-49 中城市的计算机网络必须包含纽约与丹佛之间的直接连接。那么应当包含哪些其他的连接，使得在每两个计算机中心之间都存在连接，并且费用最少？
21. 求图 8-51 带权图中包含边 $\{e, i\}$ 和 $\{g, k\}$ 的总权最小的生成树。
22. 描述一个算法，它求连通带权无向简单图中包含所规定的一组边的权最小的生成树。
23. 用伪代码表达练习 22 中设计的算法。

索林算法从连通带权简单图 $G=(V,E)$这样产生最小生成树：相继添加成组的边。假定对 V 中顶点进行了排序。这样就产生边的一个顺序，其中若 u_0 先于 u_1，或者若 $u_0=u_1$ 并且 v_0 先于 v_1，则 $\{u_0, v_0\}$ 先于 $\{u_1, v_1\}$。这个算法首先同时选择每个顶点关联的权最小的边。在权相等的情形下选择按上述顺序的第一条边。这样就产生出一个没有简单回路的图，即一些树组成的一个森林（练习 24 要求证明这个事实）。其次，对森林中的每棵树，同时选择在该树中一个顶点与在一棵不同树中顶点之间最短的边。同样在边相等的情形下选择按上述顺序的第一条边。（这样就产生出一个没有简单回路的图，它包含着比在这一步之前出现的更少的树；见练习 24。）继续进行同时添加连接树的边的过程，直到选择了 $n-1$ 条边为止。在这个阶段就构造了一棵最小生成树。

*24. 证明：在索林算法的每个阶段上边的添加都产生一个森林。
25. 用索林算法产生下图中所显示的带权图的最小生成树。
 a）图 8-49 b）图 8-51
*26. 用伪代码表达索林算法。
**27. 证明：索林算法产生连通无向带权图中的最小生成树。
*28. 证明：索林算法的第一步产生至少包含$\lceil n/2\rceil$条边的森林。
*29. 证明：若在索林算法的某个中间步骤中存在 r 棵树，则算法的下一次迭代至少添加$\lceil r/2\rceil$条边。
*30. 证明：在完成索林算法的第一步并且执行索林算法的第二步 $k-1$ 次之后，还剩下不超

过$\lceil n/2^k \rceil$棵树。

*31. 证明：从带有 n 个顶点的连通无向带权图产生一棵最小生成树，索林算法至多需要 $\log n$ 次迭代。

32. 证明：克鲁斯卡尔算法产生最小生成树。

关键术语和结果

术语

树：没有简单回路的连通无向图

森林：没有简单回路的无向图

根树：具有一个称为根的规定顶点的有向图，从根到任意其他顶点有唯一的通路。

子树：本身也是一棵树的树的子图

根树里 v 的父亲：使得(u,v)是根树的一条边的顶点 u

根树里顶点 v 的儿子：以 v 作为父亲的任何顶点

根树里顶点 v 的兄弟：与 v 具有相同父亲的顶点

根树里顶点 v 的祖先：在从根到 v 的通路上的任何顶点

根树里顶点 v 的后代：以 v 作为祖先的任何顶点

内点：具有儿子的顶点

树叶：没有儿子的顶点

顶点的层数：从根到这个顶点的通路的长度

树的高度：树里顶点的最大层数

m 元树：具有每个内点都有不超过 m 个儿子的性质的树

满 m 元树：具有每个内点都有恰好 m 个儿子的性质的树

二叉树：满足 $m=2$ 的 m 元树（可以指定每个儿子作为父亲的左儿子或右儿子）

有序树：在其中对每个内点的儿子都线性地排序的树

平衡树：在其中每个顶点都是在 h 层或 $h-1$ 层上的树，其中 h 是这棵树的高度

二叉搜索树：这样的二叉树，在其中以项对顶点进行标记，使得一个顶点的标记大于这个顶点的左子树里所有顶点的标记，并且小于这个顶点的右子树里所有顶点的标记

决策树：这样的根树，在其中每个顶点表示一次决策的可能结果，而树叶表示可能的解

前缀码：具有这样一种性质的编码，一个字符的编码永远不是另外一个字符的编码的前缀

树的遍历：树的顶点的列表

前序遍历：递归定义的有序根树的顶点列表，规定列出根，接着列出第一棵子树，接着以从左到右的顺序列出其余子树

中序遍历：递归定义的有序根树的顶点列表，规定列出第一棵子树，接着列出根，接着以从左到右的出现顺序列出其余子树

后序遍历：递归定义的有序根树的顶点列表，规定以从左到右的顺序列出各子树，接着列出根

中缀记法：从表示表达式（包括全套括号）的二叉树的中序遍历所获得的表达式形式

前缀（或波兰）记法：从表示表达式的二叉树的前序遍历所获得的表达式形式

后缀（或逆波兰）记法：从表示表达式的二叉树的后序遍历所获得的表达式形式

排序问题：以升序排列项目列表的问题

生成树：包含图的所有顶点的树

最小生成树：带最小可能的边的权之和的生成树

贪心算法：通过在每步上做出最优选择的最优化算法

结果

一个图是树，当且仅当在它的任何两个顶点之间都存在唯一简单通路。

带有 n 个顶点的树具有 $n-1$ 条边。

带有 i 个内点的满 m 元树具有 $mi+1$ 个顶点。

满 m 元树的顶点数、树叶数和内点数之间的关系（见 8.1 节里定理 4。）

在高度为 h 的满 m 元树里至多有 m^h 个树叶。

若 m 元树有 l 个树叶，则它的高度至少是 $\lceil \log_m l \rceil$。若这树也是满的和平衡的，则它的高度就是 $\lceil \log_m l \rceil$。

冒泡排序：在每遍里交换顺序颠倒的相邻项目，使用多遍来完成的排序过程

归并排序：通过相继地合并原来列表的成对子列表来完成的排序过程

深度优先搜索，或回溯：构造生成树的过程，通过添加形成通路的边，直到不可能这样做为止，然后沿这条通路往回移动，直到找到可以形成新的通路的顶点为止

宽度优先搜索：构造生成树的过程，通过相继添加与上次添加的边相关联的所有边，除非形成简单回路

普林算法：产生带权图里最小生成树的过程，通过相继添加与已经在树里的顶点相关联的所有边中权最小的边，使得没有边在添加时会产生简单回路

克鲁斯卡尔算法：产生带权图里最小生成树的过程，通过相继添加还不在树里的权最小的边，使得没有边在添加时会产生简单回路

复习题

1. a) 定义树。　　b) 定义森林。
2. 在树的顶点之间能否有两条不同的简单通路？
3. 给出如何在建模中使用树的至少三个例子。
4. a) 定义根树和这样的树的根。
   b) 定义根树里顶点的父亲和顶点的儿子。
   c) 什么是根树里的内点、树叶和子树？
   d) 画出至少带 10 个顶点的根树，其中每个顶点的度都不超过 3。指出树根、每个顶点的父亲、每个顶点的儿子、内点和树叶。
5. a) 带 n 个顶点的树有多少条边？
   b) 你需要知道什么值便能确定带有 n 个顶点的森林的边数？
6. a) 定义满 m 元树
   b) 若满 m 元树有 i 个内点，则它有多少个顶点？这树有多少个树叶？
7. a) 什么是根树的高度？
   b) 什么是平衡树？

c）高度为 h 的 m 元树可以有多少个树叶？

8. a）什么是二叉搜索树？

b）描述构造二叉搜索树的算法。

c）构造单词 *vireo*，*warbler*，*egret*，*grosbeak*，*nuthatch* 和 *kingfisher* 的二叉搜索树。

9. a）什么是前缀码？

b）二叉树如何表示前缀码？

10. a）定义前序遍历、中序遍历和后序遍历。

b）给出你所选择的至少带 12 个顶点的二叉树的前序遍历、中序遍历和后序遍历。

11. a）解释一下如何用前序遍历、中序遍历和后序遍历来求算术表达式的前缀形式、中缀形式和后缀形式。

b）画出表示 $((x-3)+((x/4)+(x-y)\uparrow 3))$ 的有序根树。

c）求在 b）里的表达式的前缀和后缀形式。

12. 证明：排序算法所使用的比较次数至少是 $\lceil \log n! \rceil$。

13. a）描述冒泡排序算法。

b）用冒泡排序算法以升序来排列表 5，2，4，1，3。

c）给出对冒泡排序算所使用的比较次数的大 O 估计。

14. a）描述归并排序算法。

b）用归并排序算法以升序来排列表 4，10，1，5，3，8，7，2，6，9。

c）给出对归并排序所使用的比较次数的大 O 估计。

15. a）什么是简单图的生成树？

b）哪些简单图具有生成树？

c）描述至少两个需要求出简单图的生成树的不同应用。

16. a）描述求简单图里生成树的两个不同算法。

b）用你所选择的至少带 8 个顶点和 15 条边的图，来解释你在 a）中所描述的两个算法是如何用来求简单图的生成树的。

17. a）解释如何用回溯来确定能否用 n 种颜色来着色简单图。

b）用例子说明如何用回溯来证明：色数等于 4 的图不能用三种颜色来着色，但是可以用四种颜色来着色。

18. a）什么是连通带权图的最小生成树？

b）描述至少两个需要求出连通带权图的最小生成树的不同应用。

19. a）描述求最小生成树的普林算法和克鲁斯卡尔算法。

b）用至少带 8 个顶点和 15 条边的图，来解释克鲁斯卡尔算法和普林算法是如何用来求最小生成树的。

补充练习

*1. 证明：简单图是树，当且仅当它不包含简单回路，并且添加连接两个不相邻顶点的一条边，就产生恰好有两条回路的新图（其中不认为包含相同的边的回路是不同的）。

*2. 有多少种非同构的带 6 个顶点的根树？

3. 证明：每一个至少有一条边的树都至少有两个悬挂点。

4. 证明：有 $n-1$ 个悬挂点的带有 n 个顶点的树必然同构于 $K_{1,n-1}$。
5. 带有 n 个顶点的树的顶点的度之和是什么？
*6. 假定 d_1，d_2，…，d_n 是和为 $2n-2$ 的 n 个正整数。证明：存在一个带有 n 个顶点的树，使得这些顶点的度为 d_1，d_2，…，d_n。
7. 证明：每个树都是平面性图。
8. 证明：每个树都是偶图。
9. 证明：每个森林都可以用两种颜色来着色。

k 度 *B* 树是一棵根树，它的所有树叶都是在同一层上，它的根具有有至少两个和至多 k 个儿子，除非根就是树叶，并且除根外的每个内点有至少 $\lceil k/2 \rceil$ 个但不超过 k 个儿子。当计算机文件用 *B* 树来表示时，就可以有效地访问这些文件。

10. 画出三种不同的高度为 4 的 3 度 *B* 树。
*11. 给出高度为 h 的 k 度 *B* 树里树叶数的上界和下界。
*12. 给出有 n 个树叶的 k 度 *B* 树的高度的上界和下界。

若根树 T 满足下面的递归定义，则称它为 S_k 树。若它只有一个顶点，则它是 S_0 树。对 $k>0$ 来说，通过两个 S_{k-1} 来建立 S_k，把一个 S_{k-1} 树的根作为 S_k 树的根，把另外一个 S_{k-1} 树的根作为第一个 S_k 树的根的儿子。

13. 对于 $k=0$，1，2，3，4 画出一个 S_k 树。
14. 证明：S_k 树有 2^k 个顶点并且在 k 层上有唯一一个顶点。在 k 层上的这个顶点称为*把柄*。
*15. 假定 T 是带有把柄 v 的 S_k 树。证明 T 可以这样从不相交的树 T_0，T_1，…，T_{k-1} 获得（其中 v 不在这些树的任何一个里，对 $i=0$，1，…，$k-1$ 来说 T_i 是 S_i 树）：对 $i=0$，1，…，$k-2$，连接 v 到 v_0 并且连接 r_i 到 r_{i+1}。

有序根树在*层数顺序*下的顶点列表从根开始，接着是从左到右在 1 层上的顶点，接着是从左到右在 2 层上的顶点，依次类推。

*16. 列出 8.3 节图 8-22 和图 8-28 里的有序根树在层数顺序下的顶点列表。
17. 设计列出有序根树在层数顺序下的顶点列表的算法。
*18. 设计确定一组通用地址能否成为根树的树叶地址的算法。
19. 设计从树叶的通用地址来构造根树的算法。

*插入排序*是这样进行的：一次考虑一个表中元素，从第二个元素开始做。每个元素都与表中前面的元素进行比较，前面的元素已经排成了正确的顺序，把这个元素插在前面元素的正确位置，这个位置原来的元素及其右边的所有元素都向右移动一个位置。

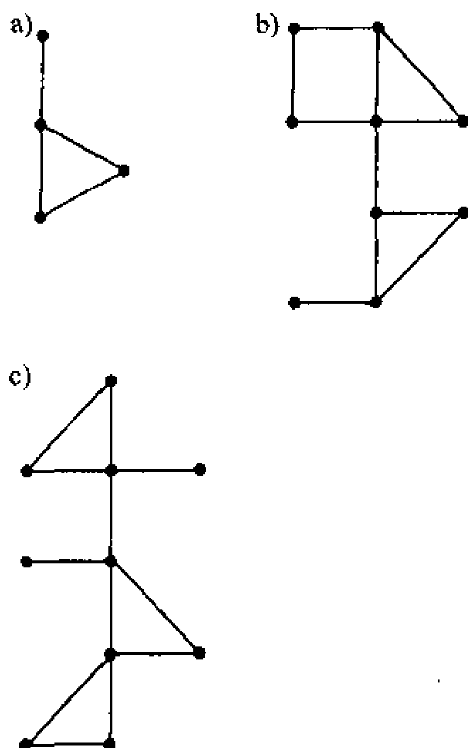
20. 用插入排序对表 3，2，4，5，1 排序。
21. 用伪代码写出插入排序。
22. 确定插入排序就所用比较次数而言的最坏情形复杂性。
23. 假定 e 是简单图里与悬挂点关联的一条边。证明：e 必然是在任何的生成树里。

图的*割集*是这样一些边的集合，删除这些边就会产生一个子图，这个子图的连通分支比原来的图要多，但是这些边的任何真子集都没有这个性质。

24. 证明：图的割集必然与这个图的任何生成树都有至少一条公共边。

仙人掌图是连通图，其中没有边是在多于一条的简单回路上，这些简单回路不经过除了起点以外的任何顶点超过一次，或者不在除了终点以外的其他地方经过起点（其中不认为由相同的边组成的两个回路是不同的）。

25. 下面的图中哪些是仙人掌图？



26. 树是否必然是仙人掌图？

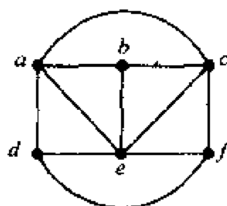
27. 证明：若在树中添加一条回路，它包含一些起止于树中一个顶点的新边，则形成一个仙人掌图。

*28. 证明：若在连通图中，每条不经过除起点以外的任何顶点超过一次的回路都包含奇数条边，则这个图必然是仙人掌图。

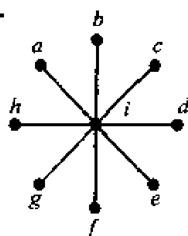
简单图 G 的**限制度数生成树**是具有这样性质的生成树，在这个树里顶点的度不能超过某个规定的界限。在运输系统的模型里，其中在交叉路口处的道路数目是有限的，在通信网络的模型里，其中进入一个结点的连接数目是有限的，这两种模型的限制度数生成树是有用的。

在练习 29～31 中，求所给的图的限制度数生成树，其中每个顶点的度都小于或等于 3，或者证明不存在这样的生成树。

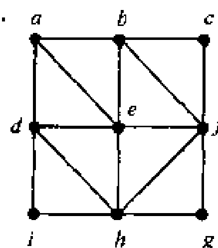
29.



30.

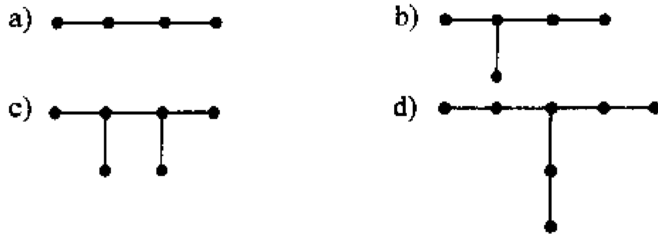


31.



32. 证明：简单图的限制度数生成树，其中每个顶点的度都不超过 2，是由该图中的单独一条哈密顿回路所组成的。

33. 若可以用整数 1，2，…，n 来标记带有 n 个顶点的树的顶点，使得相邻顶点的标记之差的绝对值全都是不同的，则这棵树称为*优美的*。证明：下面的树都是优美的。



*毛虫图*是含有一条简单通路的树，树中不包含在这条通路中的每个顶点都与这条通路中的一个顶点相邻。

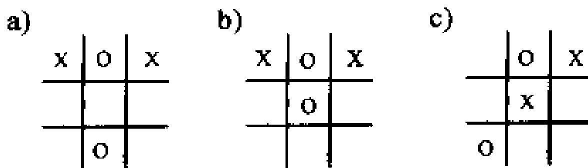
34. 在练习 33 中哪些图是毛虫图？

35. 带六个顶点的互不同的构毛虫图有多少种？

*36. a) 证明或反驳：其边形成一条简单通路的所有树都是优美的。

**b) 证明或反驳：所有的毛虫图都是优美的。

37. 假定一种跳棋游戏的前四步如下所示。解释一下如何用树来表示这种游戏的可能的后继移动。若使用 X 的选手先走，则这个选手是否有必胜的策略？（译者注：三子横竖斜连成一线即胜。）

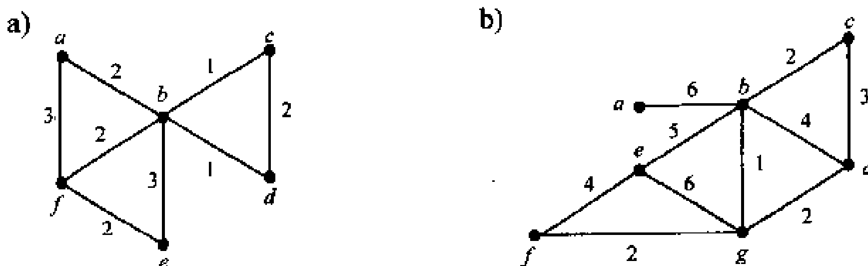


38. 三对夫妇到达一条河流的岸边。每个妻子都是嫉妒的，当她的丈夫与其他的妻子（或其他人）在一起而她不在场时，她就不信任她的丈夫。用一条只能装载不超过两个人的船，如何能使 6 个人渡河到对岸，没有丈夫与他的妻子之外的女人单独相处？使用图论模型。

*39. 假定 e 是带权图中与顶点 v 关联的一条边，e 的权不超过与顶点 v 关联的任何其他边的权。证明：存在一棵包含这条边的最小生成树。

*40. 证明：若在带权图中没有两条边具有相同的权，则在每个最小生成树里都包含着与顶点 v 关联的权最小的边。

41. 求下面每个图的最小生成树，其中在生成树中每个顶点的度数都不超过 2。



计算机题目

编写带有下列输入与输出的程序。

1. 给定无向简单图的相邻矩阵，确定这个图是不是树。
2. 给定根树的相邻矩阵和这棵树里的一个顶点，求出这个顶点的父亲、儿子、祖先、后代和层数。
3. 给定根树的边的列表和这棵树里的一个顶点，求出这个顶点的父亲、儿子、祖先、后代和层数。
4. 给定项的列表，构造包含这些项的二叉搜索树。
5. 给定二叉搜索树和一个项，在这个二叉搜索树里求出这个项的位置或添加这个项。
6. 给定有序根树的边的有序列表，求出它的边的通用地址。
7. 给定有序根树的边的有序列表，以前序、中序和后序列出它的顶点。
8. 给定前缀形式的算术表达式，求它的值。
9. 给定后缀形式的算术表达式，求它的值。
10. 给定一组 n 个整数，用冒泡排序来对它们排序。
11. 给定两个排序的整数列表，把它们合并成一个排序的列表，跟踪所使用的比较次数。
12. 给定一组 n 个整数，用归并排序来对它们排序。
13. 给定连通无向简单图的相邻矩阵，用深度优先搜索找出这个图的生成树。
14. 给定连通无向简单图的相邻矩阵，用宽度优先搜索找出这个图的生成树。
15. 给定一组正整数和一个正整数 N，利用回溯求这些整数的其和为 N 的子集合。

*16. 给定无向简单图的相邻矩阵，若有可能，就利用回溯以 3 种颜色着色这个图。

*17. 给定一个正整数 n，利用回溯来解决 n 皇后问题。

18. 给定带权无向连通图的边的列表和它们的权，用普林算法求这个图的最小生成树。
19. 给定带权无向连通图的边的列表和它们的权，用克鲁斯卡尔算法求这个图的最小生成树。

计算和研究

利用计算程序或你已经编写的程序来做下面的练习。

1. 显示所有的带有 6 个顶点的树。
2. 显示全部的互不同构的带有 7 个顶点的树。

*3. 根据英文字母在普通英文资料中出现的频度，构造一种字母的霍夫曼编码。

4. 对 $n=1, 2, 3, 4, 5, 6$ 计算 K_n 的不同生成树的个数。推测 n 为正整数时计算这种生成树个数的一个公式。
5. 对 $n=100$，1 000 和 10 000，对比对 n 个元素的列表进行选择排序、插入排序、归并排序和快速排序所需的比较次数，其中列表元素是随机选择的正整数。
6. 在 $n\times n$ 棋盘上安置 n 个皇后，使得对于任何不超过 10 的正整数 n，两个皇后不能彼此攻击。计算作出这样安置的不同方式的数目。

*7. 求美国 50 个州的首府城市连接图的最小生成树，相互连接的各边的权是城市之间的

距离。

8. 对 4×4 棋盘上的棋赛画出完全比赛树。

写作题目

对下列问题，用课外资料写成短文。

1. 说明凯莱（Arthur Cayley）如何用树来罗列碳水化合物特定类型的数目。
2. 定义 *AVL* 树（也称为高平衡树）。说明 *AVL* 树如何以及为何用在各种不同的算法中。
3. 定义四叉树，说明如何用它来表示图像。说明如何通过对相应四叉树的操纵对图像进行旋转、缩放和转换。
4. 定义一个堆，说明怎样把树转变成堆，为什么堆在排序中是很有用的？
5. 描述基于字母频度（包括霍夫曼编码）的数据压缩算法，以及基于字母块频度的相关算法。
6. 讨论如何把树用于建立游戏的模型，并提出游戏获胜的策略。说明如何用树对一种游戏进行研究，如 tic-tac-toe（井字9格游戏）、nim、hex 或其他游戏。一定要讨论 $\alpha-\beta$ 剪枝法。
7. 定义称为树格网的图。说明如何将这种图用于超大型系统集成和并行计算的应用中。
8. 讨论在 IP 组播中用于避免在路由器之间产生环的算法。
9. 描述求一个图的最小生成树的一种算法，生成树中的任何顶点的度不超过一固定的常数 k。
10. 就算法的复杂性及何时使用它们来对比某些最重要的排序算法。
11. 讨论构造最小生成树算法的历史和起源。
12. 描述产生随机树的算法。

第9章 布尔代数

计算机和其他电子设备中的电路都有输入和输出，输入是0或1，输出也是0或1。电路可以用任何具有两个不同状态的基本元件来构造，开关和光学装置都是这样的元件。开关可能处于开或关的位置，光学装置可能是点亮或未点亮的。1854年，乔治·布尔（George Boole）在《*The Laws of Thought*》一书中第一次给出了逻辑的基本规则。1938年，克劳德·香农（ClaudeShannon）[⊖]揭示了怎样用逻辑的基本规则来设计电路，这些基本规则形成了布尔代数的基础。本章将逐步展开布尔代数性质的讨论。电路的操作是由布尔函数定义的，布尔函数对输入的每个集合都指出其输出的值。构造电路的第一步是用由布尔代数的基本运算构造出的表达式来表示布尔函数。我们将提供一个算法来产生这些表达式，所得到的表达式可能包含一些冗余运算。在本章的后面，我们将描述一些求表达式的方法，用这些方法求得的表达式所包含的和与积的个数是表示一个布尔函数所需数量中最少的。将要展开讨论的这些方法称为卡诺（Karnaugh）图方法和奎因-莫可拉斯基（Quine-McCluskv）方法，它们对于设计有效的电路十分重要。

9.1 布尔函数

9.1.1 引言

布尔代数提供的是集合 {0，1} 上的运算和规则，这个集合及布尔代数的规则还可以用来研究电子和光学开关。我们将来用得最多的三个布尔代数运算是补、布尔和与布尔积。元素的补以上划线标记，其定义为：$\overline{0}=1$，且$\overline{1}=0$。布尔和记为+或OR，它有如下值：

$$1+1=1,\qquad 1+0=1,\qquad 0+1=1,\qquad 0+0=0$$

布尔积记为·或AND，它有如下值：

$$1\cdot 1=1,\qquad 1\cdot 0=0,\qquad 0\cdot 1=0,\qquad 0\cdot 0=0$$

在不引起混淆的情形下，可以删去，就像在写代数积时那样。除非使用括号，布尔运算的优先级规则是：首先计算所有补，接着是布尔积，然后是布尔和，如例1所示。

⊖ 克劳德·艾尔伍德·香农（Claude Elwood Shannon）于1916年生于密歇根州的盖劳得（Gaylord），1936年毕业于密歇根大学，之后在麻省理工学院继续学习。在麻省理工学院，他得到了一份维护微分分析器的工作，这种机器是由轴和齿轮构成的机械计算装置，是由他的硕士论文导师文那瓦·布什（Vannevar Bush）构造的。香农的硕士论文写于1936年，研究微分分析器的逻辑方面。该论文第一次提出了布尔代数在开关电路设计中的应用，它也可能是20世纪最著名的硕士论文。香农于1940年从麻省理工学院获得博士学位，并于1940年加入贝尔实验室，在那里他从事数据有效传输方面的工作，他是首批用数位表示信息的人中的一个。在贝尔实验室，他还从事于确定电话线所能承载的流量方面的工作。香农对信息论作出了许多十分重要的贡献。在20世纪50年代的早期，他是人工智能的奠基人之一。他在1956年进入麻省理工学院继续从事信息论研究。

香农有着不同于常规的一面。他被认为是以火箭为动力的塑料玩具飞盘的发明者。他曾在贝尔实验室的门厅里骑着单轮脚踏车，并同时耍着4个球，且因此而闻名。香农50岁时就退休了，但在其后的十多年中还零星地发表文章。现在他正享受生活并从事一些宠物计划，如建造一个用于跳跃的电动高跷杖。香农的一句有意思的语录是："我可以想象，我们将成为机器人而狗将成为人的时刻将会到来，我为机器鼓气加油（I visualize a time when we will be to robots what dogs are to humans. And I am rooting for the machines)。"这句话于1987年发表在《*Omni Magazine*》杂志上。

例 1 计算$1\cdot 0+\overline{(0+1)}$的值。

解 根据补、布尔和与布尔积的定义得

$$\begin{aligned}1\cdot 0+\overline{(0+1)}&=0+\overline{1}\\&=0+0\\&=0\end{aligned}$$

■

补、布尔和与布尔积分别为应于逻辑算子¬、∨和∧，且0对应于F（假），1对应于T（真）。有关布尔代数的结果可以直接翻译成关于命题的结果。相反地，关于命题的结果也能被翻译成关于布尔代数的命题。

9.1.2 布尔表达式和布尔函数

设$B=\{0,1\}$。如果变元x仅从B中取值，则称该变元为布尔变元。从B^n到B的函数被称为n度布尔函数[⊖]，其中B^n是集合$\{(x_1,x_2,\cdots,x_n)\mid x_i\in B,1\leqslant i\leqslant n\}$。布尔函数的值通常用表来表示。例如，对于如下布尔函数$F(x,y)$：当$x=1$且$y=0$时，它的值为1；当x和y取其他值时，它的值都为0。则$F(x,y)$可由表9-1表示。

表 9-1

x	y	$F(x,y)$
1	1	0
1	0	1
0	1	0
0	0	0

布尔函数也可用由变元和布尔运算构成的表达式来表示。关于变元x_1，x_2，…，x_n的布尔表达式可以递归地定义如下：

1）0，1，x_1，x_2，…，x_n是布尔表达式。

2）如果E_1和E_2是布尔表达式，则$\overline{E_1}$，(E_1E_2)和(E_1+E_2)是布尔表达式。

每个布尔表达式都表示一个布尔函数，此函数的值是通过在表达式中用0和1替换变元得到的。在9.2节中我们将介绍布尔函数是怎样由布尔表达式表示的。

例 2 计算由$F(x,y,z)=xy+\overline{z}$表示的布尔函数的值。

解 这个函数的值由表9-2表示。 ■

表 9-2

x	y	z	xy	$\overline{z}$	$F(x,y,z)=xy+\overline{z}$
1	1	1	1	0	1
1	1	0	1	1	1
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	0	0	0
0	0	0	0	1	1

布尔函数F和G是相等的，当且仅当$F(b_1,b_2,\cdots,b_n)=G(b_1,b_2,\cdots,b_n)$对$B$中的任意$b_1$，$b_2$，…，$b_n$成立。表示同一个函数的不同布尔表达式被称为是等价的。例如，布尔表达式xy，$xy+0$和$xy\cdot 1$都是等价的。布尔函数F的补函数是$\overline{F}$，此处$\overline{F}(x_1,\cdots,x_n)=\overline{F(x_1,\cdots x_n)}$。设$F$和$G$是$n$度的布尔函数，函数的布尔和$F+G$与布尔积$FG$分别定义为

$$(F+G)(x_1,\cdots,x_n)=F(x_1,\cdots,x_n)+G(x_1,\cdots,x_n)$$

⊖ 原文为 Boolean function of degree n，它常称为n元布尔函数。——译者注

$$(FG)(x_1,\cdots,x_n)=F(x_1,\cdots,x_n)G(x_1,\cdots,x_n)$$

2 度布尔函数是从一个有 4 个元素的集合到 B 的函数，这 4 个元素是 $B=\{0,1\}$ 中元素构成的元素对。B 是个有 2 个元素的集合，因而有 16 个不同的 2 度布尔函数。在表 9-3 中，我们列出了这 16 个不同的 2 度布尔函数的值，这 16 个不同的 2 度布尔函数被记为 F_1，F_2，…，F_{16}。

表 9-3　2 度布尔函数

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F^{14}	F_{15}	F_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

例 3　有多少个不同的 n 度布尔函数？

解　由计数的乘积规则知：有 2^n 个由 0 和 1 构成的不同的 n 元组。因为布尔函数就是对这些 2^n 个 n 元组中的每一个进行赋值，因而乘积规则表明了有 2^{2^n} 个不同的 n 度布尔函数。

表 9-4 列出了 1～6 度不同布尔函数的数量，这样的函数的数量增长非常快。

表 9-4　n 度布尔函数的数量

度　数	数　量
1	4
2	16
3	256
4	65 536
5	4 294 967 296
6	18 446 744 073 709 551 616



9.1.3　布尔代数中的恒等式

布尔代数有许多恒等式，表 9-5 列出了其中最重要的部分。（读者应将这些恒等式与第 1.2 节的表 1-12 中的逻辑等价式以及第 1.5 节的表 1-17 中的集合恒等式进行比较，所有这些都是一个更抽象结构中恒等式集合的特殊情形。）这些恒等式对于电路设计的简化特别有用。表 9-5 中的每个恒等式都可以用表来证明。下面的例子就以这种方法证明了一个分配律，其余性质的证明留作练习。

表 9-5　布尔恒等式

恒　等　式	名　称
$\bar{\bar{x}}=x$	双重补律
$x+x=x$ $x\cdot x=x$	幂等律
$x+0=x$ $x\cdot 1=x$	同一律
$x+1=1$ $x\cdot 0=0$	支配律
$x+y=y+x$ $xy=yx$	交换律

（续）

恒等式	名称
$x+(y+z)=(x+y)+z$ $x(yz)=(xy)x$	结合律
$x+yz=(x+y)(x+z)$ $x(y+z)=xy+xz$	分配律
$\overline{(xy)}=\overline{x}+\overline{y}$ $\overline{(x+y)}=\overline{x}\overline{y}$	德摩根律

例 4 证明分配律 $x(y+z)=xy+xz$ 是正确的。

解 表 9-6 表示了此恒等式的验证。这个恒等式成立是因为此表的最后两列相同。 ■

表 9-6

x	y	z	$y+z$	xy	xz	$x(y+z)$	$xy+xz$
1	1	1	1	1	1	1	1
1	1	0	1	1	0	1	1
1	0	1	1	0	1	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

例 5 用布尔代数的恒等式证明吸收律 $x(x+y)=x$。（之所以称为吸收律是因为将 $x+y$ 吸收进 x 而保持 x 不动。）

解 推导此恒等式的步骤及每步使用的定律如下：

$$
\begin{aligned}
x(x+y) &= (x+0)(x+y) && \text{布尔和的同一律} \\
&= x+0\cdot y && \text{布尔和对布尔积的分配律} \\
&= x+y\cdot 0 && \text{布尔积的交换律} \\
&= x+0 && \text{布尔积的支配律} \\
&= x && \text{布尔和的同一律}
\end{aligned}
$$
■

9.1.4 对偶性

表 9-5 中的恒等式都是成对出现的（除了双重补律）。为解释每一对中的两个恒等式间的关系，我们使用“对偶”这个概念。一个布尔表达式的对偶可如下得到：交换布尔和与布尔积，且交换 0 与 1。

例 6 写出 $x(y+0)$ 和 $\overline{x}\cdot 1+(\overline{y}+z)$ 的对偶。

解 在这两个表达式中交换符号 + 和 · 以及 0 和 1 就产生了它们的对偶。这两个对偶分别是 $x+(y\cdot 1)$ 和 $(\overline{x}+0)(\overline{y}z)$。 ■

布尔表达式所表示的布尔函数 F 的对偶是由这个表达式的对偶所表示的函数，这个对偶函数记为 F^d，它不依赖于表示 F 的那个特定的布尔表达式。对于由布尔表达式表示的函数的恒等式，当取恒等式两边的函数的对偶时，等式仍然成立（原因参见练习 22）。此结果叫做*对偶性原理*，它对于获得新的恒等式十分有用。

例 7　通过取对偶的方法，由例 5 中的吸收律 $x(x+y)=x$ 构造一个恒等式。

解　取此恒等式两边的对偶，得到恒等式 $x+xy=x$，它也被称为吸收律。 ■

9.1.5　布尔代数的抽象定义

本节中我们一直专注于布尔函数和表达式，但已得到的结果都可以翻译成关于命题的结果或关于集合的结果，因此，抽象地定义布尔代数十分有用。一旦一个特定的结构被证明是布尔代数，则所有关于布尔代数的一般结果都可应用于这个特定的结构。

布尔代数可以用多种方法定义，最常用的方法是指明运算所必须满足的性质，如下面的定义所示。

定义 1　一个布尔代数是一个集合 B，它有两个二元运算 $\vee$ 和 $\wedge$，元素 0 和 1，以及一个一元运算 $^{-}$，且对 B 中的所有元素 x、y 和 z，下列性质成立：

$$\left.\begin{array}{l} x \vee 0 = x \\ x \wedge 1 = x \end{array}\right\} \quad \text{同一律}$$

$$\left.\begin{array}{l} x \vee \bar{x} = 1 \\ x \wedge \bar{x} = 0 \end{array}\right\} \quad \text{支配律}$$

$$\left.\begin{array}{l} (x \vee y) \vee z = x \vee (y \vee z) \\ (x \wedge y) \wedge z = x \wedge (y \wedge z) \end{array}\right\} \quad \text{结合律}$$

$$\left.\begin{array}{l} x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \end{array}\right\} \quad \text{分配律}$$

使用定义 1 所给的定律可以证明许多其他的定律，例如幂等律、支配律等。（见练习 25～32。）

以前讨论过，$B=\{0,1\}$ 连同 OR、AND 运算及“补”运算满足所有这些性质。有 n 个变元的所有命题构成的集合，连同 $\vee$ 和 $\wedge$ 算子、**F** 和 **T** 及“非”算子，也满足布尔代数的所有性质，这可以从第 1.2 节中的表 1-12 中看出来。类似地，一个全集 U 的所有子集构成的集合，连同并和交运算、空集和全集及集合求补运算是一个布尔代数，这可以从第 1.5 节的表 1-17 中看出来。所以，为了建立关于布尔表达式、命题和集合的结果，我们只要证明关于抽象布尔代数的结果即可。

布尔代数也可以用第 6 章中所讨论的格的概念来定义。一个格 L 是一个偏序集，其每对元素 x、y 都有一个最小上界，记为 $\mathrm{lub}(x,y)$；也有一个最大下界，记为 $\mathrm{glb}(x,y)$。给定 L 的两个元素 x 和 y，我们可以定义 L 的两个运算 $\vee$ 和 $\wedge$ 如下：$x\vee y=\mathrm{lub}(x,y)$ 且 $x\wedge y=\mathrm{glb}(x,y)$。

要使一个格成为定义 1 所指出的一个布尔代数，它必须有两个性质。第一，它必须是可补的。为使一个格成为可补的，它必须有一个最小元素 0 和一个最大元素 1，且对格的每个

元素 x，必须存在一个元素$\overline{x}$，使得 $x \vee \overline{x}=1$ 且 $x \wedge \overline{x}=0$。第二，它必须是分配的。所谓"分配的"指的是：对于 L 中的每个 x、y 和 z，$x \vee (y \wedge z)=(x \vee y) \wedge (x \vee z)$且$x \wedge (y \vee z)=(x \wedge y) \vee (x \wedge z)$。证明可补的分配格是布尔代数放在本节末尾的练习 33 中。

练习

1. 求下列表达式的值。
   a) $1 \cdot 0$　　b) $1+1$　　c) $\overline{0} \cdot 0$　　d) $\overline{(1+0)}$
2. 求满足下列方程的布尔变元的值（如果有的话）。
   a) $x \cdot 1=0$　　b) $x+x=0$　　c) $x \cdot 1=x$　　d) $x \cdot \overline{x}=1$
3. 布尔变元 x 和 y 的什么值满足 $xy=x+y$？
4. 有多少个 7 度布尔函数？
5. 用表 9-5 中的定律证明吸收律 $x+xy=x$。
6. 证明 $F(x,y,z)=xy+xz+yz$ 取值 1，当且仅当变元 x、y 和 z 中至少有两个取值 1。
7. 证明 $x\overline{y}+y\overline{z}+\overline{x}z=\overline{x}y+\overline{y}z+x\overline{z}$。

练习 8～15 处理由 {0，1} 上的布尔和与布尔积所定义的布尔代数。

8. 验证双重补律。
9. 验证幂等律。
10. 验证同一律。
11. 验证支配律。
12. 验证交换律。
13. 验证结合律。
14. 验证表 9-5 中的第一个分配律。
15. 验证德摩根律。

布尔算子⊕的定义如下：$1 \oplus 1=0$，$1 \oplus 0=1$，$0 \oplus 1=1$，$0 \oplus 0=0$。此算子被称为"异或(XOR)"算子。

16. 化简下列表达式。
    a) $x \oplus 0$　　b) $x \oplus 1$　　c) $x \oplus x$　　d) $x \oplus \overline{x}$
17. 证明下列恒等式成立。
    a) $x \oplus y=(x+y)\overline{(xy)}$　　b) $x \oplus y=(x\overline{y})+(\overline{x}y)$
18. 证明 $x \oplus y=y \oplus x$。
19. 证明下列等式成立或不成立。
    a) $x \oplus (y \oplus z)=(x \oplus y) \oplus z$
    b) $x+(y \oplus z)=(x+y) \oplus (x+z)$
    c) $x \oplus (y+z)=(x \oplus y)+(x \oplus z)$
20. 求下列布尔表达式的对偶。
    a) $x+y$　　b) $\overline{x}\,\overline{y}$
    c) $xyz+\overline{x}\,\overline{y}\,\overline{z}$　　d) $x\overline{z}+x \cdot 0+\overline{x} \cdot 1$

*21. 设 F 是一个布尔函数，它由一个含有变元 x_1，…，x_n 的布尔表达式表示。证明 $F^d(x_1,\cdots,x_n)=\overline{F(\overline{x}_1,\cdots,\overline{x}_n)}$

*22. 设 F 和 G 是由 n 个变元的布尔表达式表示的布尔函数，$F=G$。证明 $F^d=G^d$，其中 F^d 和 G^d 分别是由表示 F 和 G 的布尔表达式的对偶表示的布尔函数。(提示：使用练习 21 的结果。)

*23. 有多少个不同的布尔函数 $F(x,y,z)$，使得对布尔变元 x、y、z 的所有值，$F(\overline{x},\overline{y},\overline{z})=F(x,y,z)$。

*24. 有多少个不同的布尔函数 $F(x,y,z)$，使得对布尔变元 x、y、z 的所有值，$F(\overline{x},y,z)=F(x,\overline{y},z)=F(x,y,\overline{z})$。

在练习 25－32 中，用定义 1 中的定律来证明所述性质对每个布尔代数成立。

25. 在布尔代数中证明幂等律 $x\vee x=x$ 和 $x\wedge x=x$ 对每个元素成立。
26. 在布尔代数中证明：每个元素 x 都有唯一的一个补 $\overline{x}$，使得 $x\vee\overline{x}=1$ 且 $x\wedge\overline{x}=0$。
27. 在布尔代数中证明：元素 0 的补是 1，反之也成立。
28. 证明双重补律在布尔代数中成立，即对每个元素 x，$\overline{\overline{x}}=x$。
29. 证明德摩根律在布尔代数中成立，即对任意元素 x 和 y，$\overline{(x\vee y)}=\overline{x}\wedge\overline{y}$ 且 $\overline{(x\wedge y)}=\overline{x}\vee\overline{y}$
30. 证明模性质在布尔代数中成立，即证明 $x\wedge(y\vee(x\wedge z))=(x\wedge y)\vee(x\wedge z)$ 且 $x\vee(y\wedge(x\vee z))=(x\vee y)\wedge(x\vee z)$。
31. 在布尔代数中证明：如果 $x\vee y=0$，则 $x=0$ 且 $y=0$；如果 $x\wedge y=1$，则 $x=1$ 且 $y=1$。
32. 在布尔代数中证明一个恒等式的对偶还是一个恒等式，其中，恒等式的对偶是如下得到的：交换 $\wedge$ 和 $\vee$ 运算，并交换元素 0 和 1。
33. 证明一个可补的分配格是一个布尔代数。

9.2 布尔函数的表示

本节将研究布尔代数的两个重要问题。第一，给定一个布尔函数，怎么才能找到表示这个布尔函数的布尔表达式？这个问题将通过证明如下结论来解决：任何一个布尔函数都可由变元及其补的布尔积的布尔和表示。这个问题的答案还说明了任意布尔函数都可用三个布尔算子表示：·、＋和－。第二，有没有一个更小的算子集合可以用来表示所有的布尔函数？我们将通过证明下列结论来解决这个问题：所有的布尔函数都可用一个算子来表示。这两个问题在电路设计中都有特殊的重要性。

9.2.1 积之和展开式

下面用例子来说明寻找表示布尔函数的布尔表达式的一个重要方法。

例 1 函数 $F(x,y,z)$ 和 $G(x,y,z)$ 如表 9-7 所示，求表示这两个函数的布尔表达式。

表 9.7

x	y	z	F	G
1	1	1	0	0
1	1	0	0	1
1	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	0

解 我们需要这样一个表达式来表示 F：当 $x=z=1$ 且 $y=0$ 时它的值为 1，否则它的值为 1。此表达式可取为 x、$\bar{y}$ 和 z 的布尔积。这个积 $x\bar{y}z$ 具有值 1 当且仅当 $x=\bar{y}=z=1$，即当且仅当 $x=z=1$ 且 $y=0$。

为表示 G，我们需要一个表达式满足：当 $x=y=1$ 且 $z=0$ 时，或当 $x=z=0$ 且 $y=1$ 时，它为 1。这样的表达式可以取为两个不同布尔积的布尔和。布尔积 $yx\bar{z}$ 具有值 1 当且仅当 $x=y=1$ 且 $z=0$；类似地，布尔积 $\bar{x}y\bar{z}$ 具有值 1 当且仅当 $x=z=0$ 且 $y=1$。这两个布尔积的布尔和 $xy\bar{z}+\bar{x}y\bar{z}$ 就表示 G，因为它具有值 1 当且仅当 $x=y=1$ 且 $z=0$，或 $x=z=0$ 且 $y=1$。 ■

例 1 说明了一个过程，用这个过程可以构造布尔表达式来表示具有给定值的函数。如果变元值的一个组合使得函数值为 1，则此组合确定了变元或其补的一个布尔积。

定义 1 布尔变元或其补称为文字。布尔变元 $x_1, x_2, \cdots, x_n$ 的小项是一个布尔积 $y_1y_2\cdots y_n$，其中 $y_i=x_i$，或 $y_i=\bar{x}_i$。因此小项是 n 个文字的积，每个文字对应于一个变元。

一个小项对且只对一个变元值的组合取值 1，更确切地讲，小项 $y_1y_2\cdots y_n$ 为 1 当且仅当每个 y_i 为 1，当且仅当 $y_i=x_i$ 时 x_i 为 1，$y_i=\bar{x}_i$ 时 x_i 为 0。

例 2 求一个小项使得：当 $x_1=x_3=0$ 且 $x_2=x_4=x_5=1$ 时，它为 1；否则为 0。

解 小项 $\bar{x}_1x_2\bar{x}_3x_4x_5$ 具有正确的值集合。 ■

通过取不同小项的布尔和，就能构造出布尔表达式，使其具有给定的值集合。特别地，小项的布尔和具有值 1 只有当和中的某个小项具有值 1 时才成立；对于变元值的其他组合，它具有值 0。因此，给定一个布尔函数，可以构造小项的布尔和使得：当此布尔函数具有值 1 时它的值为 1，当此布尔函数具有值 0 时它的值为 0。此布尔和中的小项与使得此函数值为 1 的值的组合相对应。表示布尔函数的小项的和称为此函数的*积之和展开式*或*析取范式*。

例 3 求函数 $F(x,y,z)=(x+y)\bar{z}$ 的积之和展开式。

解 第一步是计算 F 的值。这些值见表 9-8。F 的积之和展开式是三个小项的布尔和，这三个小项对应于表 9-8 的三行，它们使此函数的值为 1。从而

$$F(x,y,z)=xy\bar{z}+x\bar{y}\bar{z}+\bar{x}y\bar{z}$$ ■

表 9-8

x	y	z	$x+y$	$\bar{z}$	$(x+y)\bar{z}$
1	1	1	1	0	0
1	1	0	1	1	1
1	0	1	1	0	0
1	0	0	1	1	1
0	1	1	1	0	0
0	1	0	1	1	1
0	0	1	0	0	0
0	0	0	0	1	0

也可以通过取布尔和的布尔积来求一个布尔表达式，使其表示一个布尔函数。所得到的展开式称为这个函数的*合取范式*或*和之积展开式*，这个展开式可以通过求积之和展开式的对偶而得到。本节末尾的练习 10 描述了怎样直接求这样的展开式。

9.2.2 函数完备性

每个布尔函数都可以表示为小项的布尔和，每个小项都是布尔变元或其补的布尔积，这说明了每个布尔函数都可以用布尔运算·，+ 和 ¯ 来表示。因为每个布尔函数都可以由布尔

运算表示，我们称集合 {·，+，¯} 是函数完备的。还有没有更小的函数完备运算集合呢？如果这三个运算中的某一个能够由其余两个表示，则就还有。用德摩根律可以做到这一点。使用恒等式

$$x+y=\overline{\overline{x}\,\overline{y}}$$

可以消去所有布尔和，此恒等式可如下得到：先对第 9.1 节中表 9-5 的第二个德摩根律的两边求补，再应用双补律。这意味着 {·，¯} 是函数完备的。类似地，使用恒等式

$$xy=\overline{\overline{x}+\overline{y}}$$

可以消去所有布尔积，此恒等式可如下得到：先对第 9.1 节中表 9-5 的第一个德摩根律的两边求补，再应用双补律。因此，{+，¯} 是函数完备的。注意，{+，·} 不是函数完备的，因为用这两个运算不可能表示布尔函数$F(x)=\overline{x}$(见练习 19)。

我们已经找到了一些含有两个运算的函数完备集合，还能不能找到更小的集合——即只含一个运算的集合——它仍然是函数完备运算集合呢？这样的集合是存在的。定义运算“|”或“NAND”如下：$1|1=0$ 且 $1|0=0|1=0|0=1$。定义运算“↓”或“NOR”如下：$1\downarrow 1=1\downarrow 0=0\downarrow 1=0$ 且 $0\downarrow 0=1$。集合 {|} 和 {↓} 都是函数完备的。因为 {·，¯} 是函数完备的，故要说明 {|} 是函数完备的，只要证明两个运算·和¯都可以由运算|表示，这由下面两式完成：

$$\overline{x}=x\,|\,x$$

$$xy=(x\,|\,y)\,|\,(x\,|\,y)$$

读者应当验证这些恒等式（见练习 14)。证明 {↓} 的函数完备性留给读者（见练习 15 和 16)。

练习

1. 求布尔变元 x、y、z 或其补的布尔积，使得它具有值为 1 当且仅当
   a) $x=y=0$，$z=1$　　b) $x=0$，$y=1$，$z=0$
   c) $x=0$，$y=z=1$　　d) $x=y=z=0$
2. 求下列布尔函数的积之和展开式。
   a) $F(x,y)=\overline{x}+y$　　b) $F(x,y)=x\,\overline{y}$
   c) $F(x,y)=1$　　d) $F(x,y)=\overline{y}$
3. 求下列布尔函数的积之和展开式。
   a) $F(x,y,z)=x+y+z$　　b) $F(x,y,z)=(x+z)y$
   c) $F(x,y,z)=x$　　d) $F(x,y,z)=x\,\overline{y}$
4. 求布尔函数$F(x,y,z)$的积之和展开式，$F(x, y, z)$ 等于 1 当且仅当
   a) $x=0$　　d) $xy=0$
   c) $x+y=0$　　d) $xyz=0$
5. 求布尔函数$F(w,x,y,z)$的积之和展开式，$F(w,x,y,z)$等于 1 当且仅当 w、x、y 和 z 中有值为 1 的变元有奇数个。
6. 求布尔函数$F(x_1,x_2,x_3,x_4,x_5)$的积之和展开式，$F(x_1,x_2,x_3,x_4,x_5)$等于 1 当且仅当

x_1，x_2，x_3，x_4 和 x_5 中至少有三个变元的值为 1。

求表示布尔函数的布尔表达式的另一种方法是：构造文字之布尔和的布尔积。练习 7～11 涉及这种表示。

7. 求布尔和，它包含 x 或$\overline{x}$、y 或 $\overline{y}$以及 z 或 $\overline{z}$，使得它具有值 1 当且仅当
   a) $x=y=1$，$z=0$　　b) $x=y=z=0$　　c) $x=z=0$，$y=1$
8. 求文字之布尔和的布尔积，使得它的值为 1 当且仅当 $x=y=1$ 且 $z=0$，或者 $x=z=0$ 且 $y=1$，或者 $x=y=z=0$。[提示：利用从练习 7 的 a)、b)、c) 部分求得的布尔和的布尔积。]
9. 设布尔和为 $y_1+y_2+\cdots+y_n$，其中：$y_i=x_i$ 或 $y_i=\overline{x}_i$。证明此布尔和对且只对变元值的一个组合取 0 值，这个组合为：若 $y_i=x_i$ 则 $x_i=0$；$y_i=\overline{x_i}$ 则 $x_i=1$。这样的布尔和叫做大项。
10. 证明布尔函数可以表示为大项的布尔积。此表示称为该函数的和之积展开式或析取范式。[提示：对于使得函数值为 0 的每个变元值组合，此积都含有一个对应的大项。]
11. 求练习 3 中每个函数的和之积展开式。
12. 用运算·和¯表示下列布尔函数。
    a) $x+y+z$　　b) $x+\overline{y}\ (\overline{x}+z)$
    c) $\overline{(x+\overline{y})}$　　d) $\overline{x}(x+\overline{y}+\overline{z})$
13. 用运算 + 和¯表示练习 12 中的布尔函数。
14. 证明：
    a) $\overline{x}=x\mid x$　　b) $xy=(x\mid y)\mid(x\mid y)$　　c) $x+y=(x\mid x)\mid(y\mid y)$
15. 证明：
    a) $\overline{x}=x\downarrow x$　　b) $xy=(x\downarrow x)\downarrow(y\downarrow y)$　　c) $x+y=(x\downarrow y)\downarrow(x\downarrow y)$
16. 利用练习 15 证明 $\{\downarrow\}$ 是函数完备集。
17. 用运算 | 表示练习 3 中的布尔函数。
18. 用运算 ↓ 表示练习 3 中的布尔函数。
19. 证明运算集 {+，·} 不是函数完备的。
20. 下列运算集是否为函数完备的呢?
    a) {+，⊕}　　b) {¯，⊕}　　c) {·，⊕}

9.3 逻辑门电路

9.3.1 引言

布尔代数被用来作为电子装置的电路模型，这样装置的输入和输出都可以认为是集合{0，1}中的元素。计算机或其他的电子装置就是由许多电路构成的，电路可以根据布尔代数的规则来设计，这些规则已经在第 9.1 和 9.2 节中讨论过。电路的基本元件是所谓的门，每种类型的门实现一种布尔运算。本节将定义几种类型的门，对这些门应用布尔代数的结果，就可以设计出电路来执行各种各样的任务。在本章所讨论的电路中，输出都只与输入有关，而与电路的当前状态无关，换句话说，这些电路都没有存储能力，这样的电路叫做

组合电路或选通网络。

我们将使用三种元件来构造组合电路，第一种是反相器，它以布尔值作为输入，并产生此布尔值的补作为输出。用来表示反相器的符号如图 9-1a）所示，进入元件的输入画在左边，离开元件的输出画在右边。

第二种元件是或门，其输入是两个或两个以上的布尔值，输出是这些值的布尔和。用来表示或门的符号如图 9-1b）所示，进入元件的输入画在左边，离开元件的输出画在右边。

第三种元件是与门，其输入是两个或两个以上的布尔值，输出是这些值的布尔积。用来表示或门的符号如图 9-1c）所示，进入元件的输入画在左边，离开元件的输出画在右边。

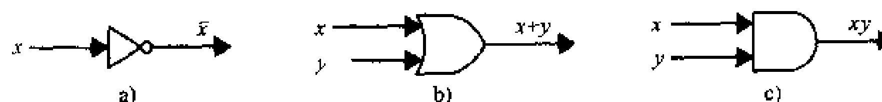


图 9-1　基本类型的门

与门和或门允许有多个输入，进入元件的输入都画在左边，离开元件的输出都画在右边。具有 n 个输入的与门和或门如图 9-2 所示。

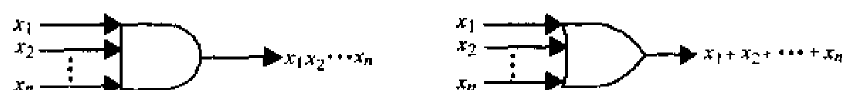


图 9-2　具有 n 个输入的门

9.3.2　门的组合

使用反相器、或门和与门的组合可以构造组合电路。在构造电路的组合时，某些门可能有公共的输入。有两种方法可以描述公共输入。一种方法是：对每个输入，将使用这个输入的门画在不同的分支上。另一种方法是：对每个门，分别指出其输入。图 9-3 说明了这两种方法，其中的门有同样的输入值。注意，一个门的输出可能被作为另一个或更多元件的输入，如图 9-3 所示。图 9-3 中的两幅图描述了输出为 $xy+\bar{x}y$ 的电路。

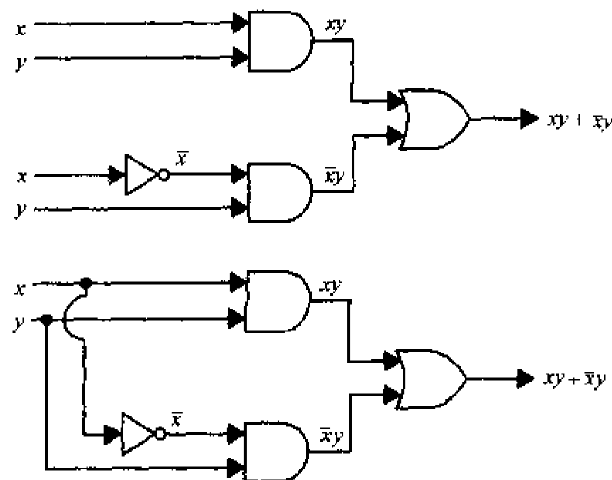


图 9-3　画相同电路的两种方法

例 1　构造产生下列输出的电路：a）$(x+y)\overline{x}$，b）$\overline{x}\,\overline{(y+\overline{z})}$，c）$(x+y+z)\overline{x}\,\overline{y}\,\overline{z}$。

解　产生这些输出的电路如图 9-4 所示。■

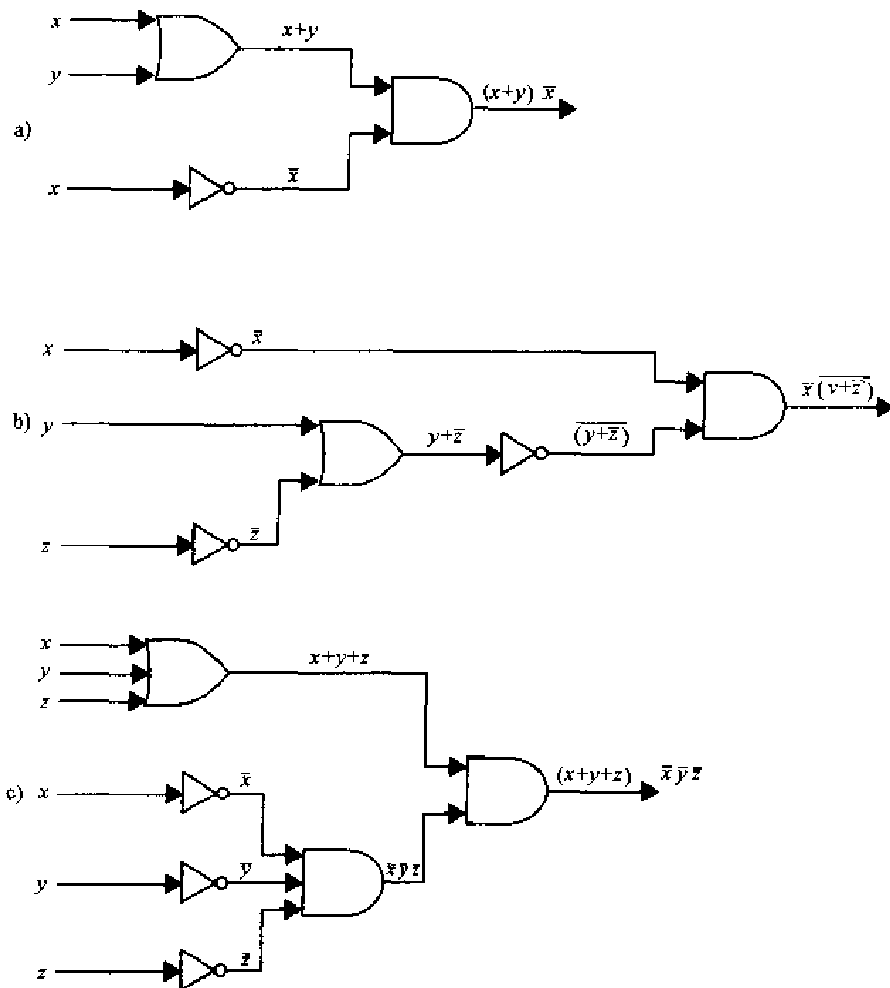


图 9-4　产生例 1 指定输出的电路

9.3.3　电路的例子

下面给出一些具有实际功能的电路。

例 2　某个组织的一切事务都由一个三人委员会决定，每个委员对提出的建议可以投赞成票或反对票。一个建议如果得到了至少两张赞成票就获通过。设计一个电路，来确定建议是否获得通过。

解　如果第一个委员投赞成票，则令 $x=1$；如果这个委员投反对票，则令 $x=0$。如果第二个委员投赞成票，则令 $y=1$；如果这个委员投反对票，则令 $y=0$。如果第三个委员投赞成票，则令 $z=1$；如果这个委员投反对票，则令 $z=0$。必须设计一个电路使得：对于输入 x、y 和 z，如果其中至少有两个为 1，则此电路产生输出 1。具有这样输出值的一个布尔函数表示是 $xy+xz+yz$（见第 9.1 节练习 6）。实现这个函数的电路如图 9-5 所示。■

例 3　有时候灯具需要由多个开关来控制，因此有必要设计这样的电路：当灯是关闭时，敲击任何一个开关都可以打开此灯；反之，当灯是打开时，敲击任何一个开关都可以关闭此灯。在有两个开关或三个开关两种情形下，设计电路来完成这个任务。

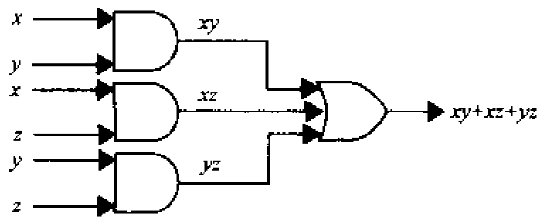


图 9-5　用于多数表决的电路

解　首先设计使用两个开关的电路来控制灯具。当第一个开关关闭时，令 $x=1$；当它打开时，令 $x=0$。当第二个开关关闭时，令 $y=1$；当它打开时，令 $y=0$。当灯是打开的时候，令 $F(x,y)=1$；当它是关闭时，令 $F(x,y)=0$。我们可以随意地假定：当两个开关都是关闭的时候，灯是打开的，即 $F(1,1)=0$。这个假定决定了 F 的所有其他值：当两个开关中有一个是打开的时候，灯变灭了，故 $F(1,0)=F(0,1)=0$；当第二个开关也被打开的时候，灯又变亮了，故 $F(0,0)=1$。表 9-9 列出了这些值。我们知道 $F(x,y)=xy+\overline{x}\,\overline{y}$。实现这个函数的电路如图 9-6 所示。

表　9-9

x	y	$F(x,y)$
1	1	1
1	0	0
0	1	0
0	0	1

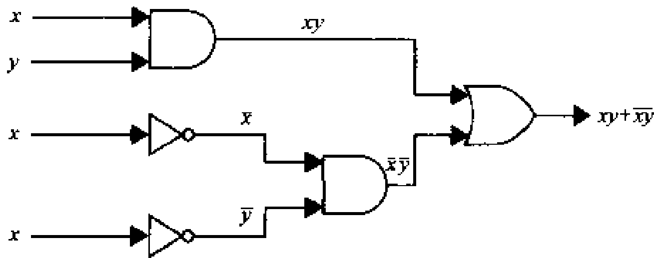


图 9-6　由两个开关控制的灯具电路

现在设计有三个开关的电路。设 x、y 和 z 是三个布尔变元，它们分别指出这三个开关是否是关闭的。当第一个开关处于关闭时，令 $x=1$；当它打开时，令 $x=0$。当第二个开关处于关闭时，令 $y=1$；当它打开时，令 $y=0$。当第三个开关处于关闭时，令 $z=1$；当它打开时，令 $z=0$。灯亮时，令 $F(x,y,z)=1$；灯不亮时，令 $F(x,y,z)=0$。当所有开关都关闭时，我们可以随意地指定灯是亮的，即 $F(1,1,1)=1$，这决定了 F 的其他值。当一个开关打开时，灯就变灭，故 $F(1,1,0)=F(1,0,1)=F(0,1,1)=0$。当又一个开关打开时，灯又变亮了，故 $F(1,0,0)=F(0,1,0)=F(0,0,1)=1$。最后当三个开关都打开时，灯又变灭了，故 $F(0,0,0)=0$。这个函数的值如表 9-10 所示。函数 F 可以表示成积之和表达式：$F(x,y,z)=xyz+x\overline{y}\,\overline{z}+\overline{x}y\overline{z}+\overline{x}\,\overline{y}z$。实现这个函数的电路如图 9-7 所示。■

表　9-10

x	y	z	$F(x,y,z)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

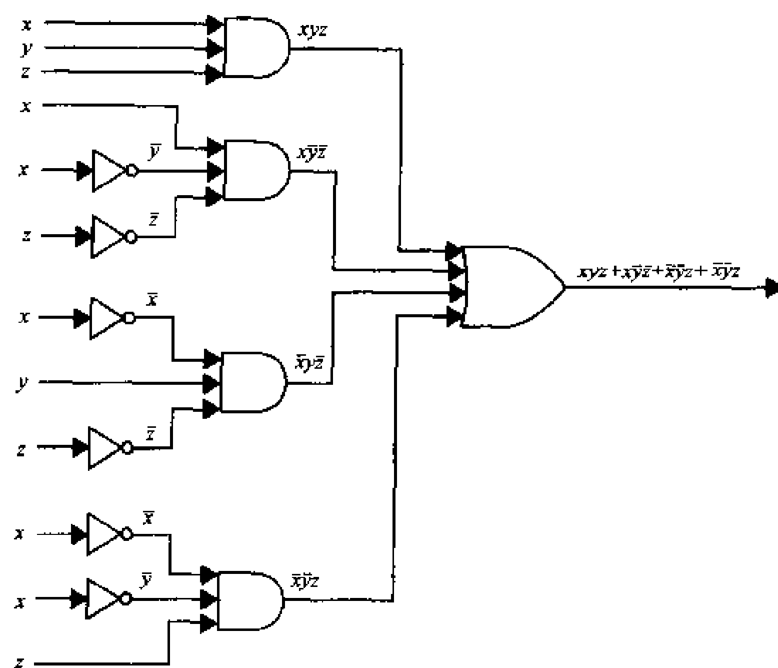


图 9-7 由三个开关控制的灯具电路

9.3.4 加法器

下面说明怎么用逻辑电路对两个正整数的二进制编码来执行加法。我们先构造一些分支电路，然后再从这些分支电路来构造加法电路。首先构造电路来计算 $x+y$，其中 x 和 y 是两个二进制数字。因为 x 和 y 的值为 0 或 1，此电路的输入就是 x 和 y。输出由两个二进制数字 s 和 c 构成，其中 s 和 c 分别是和位与进位。因为这种电路具有多个输出，故称为**多重输出电路**。又由于此电路只是将两个二进制数字相加，而没有考虑以前加法所产生的进位，所以这样的电路称为**半加器**。表 9-11 说明了半加器的输入和输出。由此表可以看出 $c=xy$，且 $s=x\bar{y}+\bar{x}y=(x+y)\overline{(xy)}$。这样图 9-8 所示的电路计算了 x 与 y 的和位 s 与进位 c。

表 9-11 半加器的输入和输出

输入		输出	
x	y	s	c
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

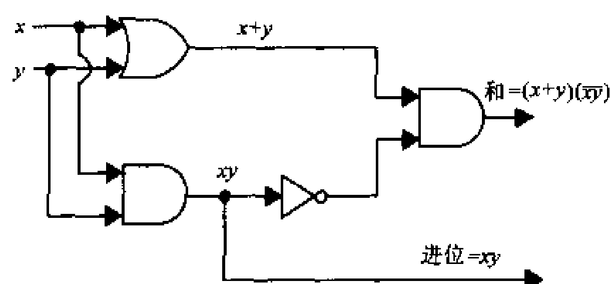


图 9-8 半加器

当两个二进制数字与一个进位相加时，我们用全加器来计算和位与进位。全加器的输入是这两个二进制数字 x 和 y 以及进位 c_i，输出是和位 s 与新的进位 c_{i+1}。全加器的输入和输出如表 9-12 所示。

全加器的两个输出——和位 s 与进位 c_{i+1}——可分别由积之和展开式 $xyc_i + x\bar{y}\bar{c}_i + \bar{x}y\bar{c}_i + \bar{x}\bar{y}c_i$ 与 $xyc_i + xy\bar{c}_i + x\bar{y}c_i + \bar{x}yc_i$ 表示。但我们并不直接构造全加器，而是使用半加器来产生所需的输出。使用半加器构造全加器的方法如图 9-9 所示。

表 9-12　全加器的输入和输出

输入			输出	
x	y	c_i	s	c_{i+1}
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

最后，图 9-10 说明了怎样用加法器和半加器来计算两个 3 位二进制整数 $(x_2x_1x_0)_2$ 与 $(y_2y_1y_0)_2$ 的和 $(s_3s_2s_1s_0)_2$。注意，和中的最高位 s_3 是由进位 c_2 产生的。

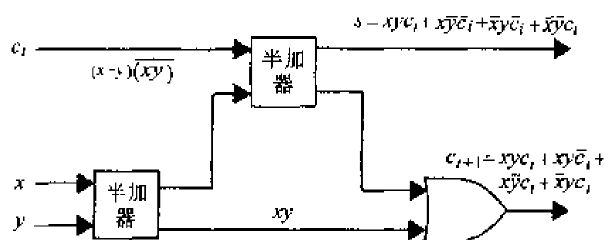


图 9-9　全加器

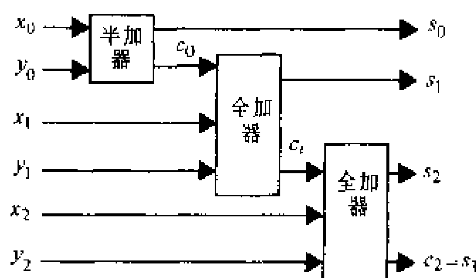
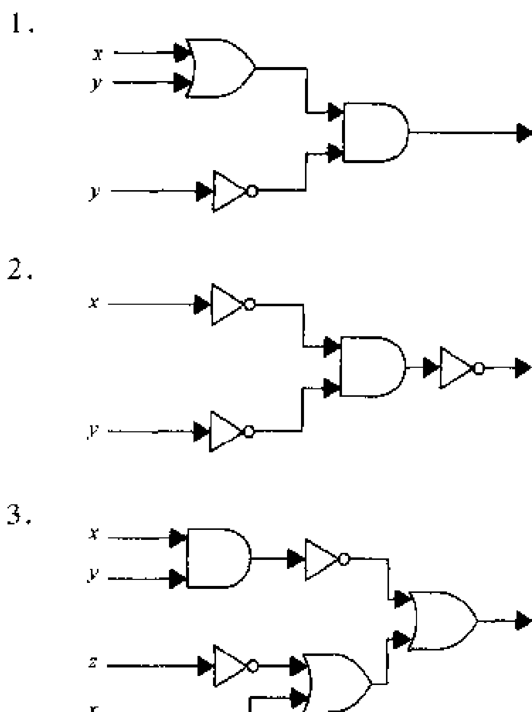
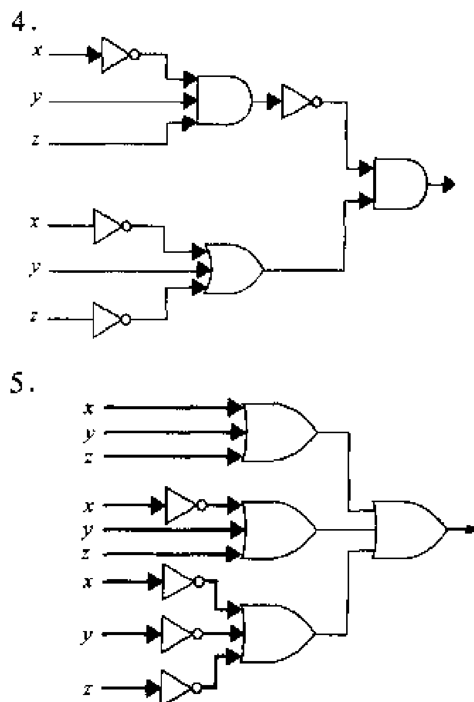


图 9-10　用全加器和半加器将两个 3 位整数相加

练习

在练习 1～5 中，求所给电路的输出。





6. 用反相器、与门和或门构造产生下列输出的电路。

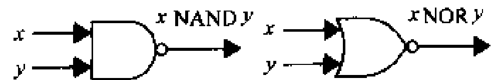
   a) $\bar{x}+y$　　b) $\overline{(x+y)x}$　　c) $xyz+\bar{x}\,\bar{y}\,\bar{z}$　　d) $\overline{(\bar{x}+z)(y+\bar{z})}$

7. 试设计一个电路来实现五个人的多数表决。
8. 试设计一个由四个开关控制的电灯混合控制器，使得当电灯在打开时，按动任意一个开关都可关闭它；或者当电灯在关闭时，按动任意一个开关都可打开它。
9. 证明可以使用全加器和半加器来计算两个 5 位二进制整数的和。
10. 一个半减器的输入是两个二进制数字，输出是差和借位。试用与门、或门和反相器构造一个半减器电路。
11. 一个全减器的输入是两个二进制数字及一个借位，输出是差和借位。试用与门、或门和反相器构造一个全减器电路。
12. 使用练习 10 和 11 中的电路计算两个 4 位二进制整数的差，其中，第一个整数大于第二个整数。

*13. 构造一个电路来比较两位二进制整数 $(x_1x_0)_2$ 和 $(y_1y_0)_2$，使得当第一个整数大于第二个时，输出 1，否则输出 0。

*14. 构造一个计算两位二进制整数 $(x_1x_0)_2$ 与 $(y_1y_0)_2$ 之积的电路，此电路应该有四个输出位。

与非门（NAND）和或非门（NOR）也是电路中常用的两个门电路，如果使用这两个门来表示电路，就没有必要使用其他类型的门了。这些门的记号如下：



*15. 使用与非门构造具有下列输出的电路。

   a) $\bar{x}$　　b) $x+y$　　c) xy　　d) $x\oplus y$

*16. 使用或非门构造具有练习 15 中输出的电路。

*17. 试用与非门构造半加器。

*18. 试用或非门构造半加器。

*多路转换器*是一种开关电路，它根据控制位的值将某组输入位输出。

19. 用与门、或门和反相器构造一个多路转换器，它的四个输入是二进制数字 x_0，x_1，x_2 和 x_3，控制位是 c_0 和 c_1。建造此电路使得 x_i 为输出，其中 i 是 2 位整数 $(c_1c_0)_2$ 的值。

9.4　电路的极小化

9.4.1　引言

组合电路的有效性依赖于门的个数及安排。在组合电路的设计过程中，首先构造一个表，对于输入可能取的每种值，此表说明对应的输出值。对于任何一个电路，总可以用“积之和展开式”找到一组逻辑门来实现这个电路。但是，积之和展开式可能包含许多不必要的项。在一个积之和展开式中，若其中的一些项只在一个变元处不一样，即在某项中此变元本身出现，而在另一项中此变元的补出现，则这些项可以合成。例如，考虑这样的电路，它输出 1 当且仅当 $x=y=z=1$，或 $x=z=1$ 且 $y=0$。此电路的积之和展开式为 $xyz+x\bar{y}z$。在

这个展开式的两个积中，只有一个变元以不同的形式出现，即 y。它们可以如下合并：

$$\begin{aligned} xyz + x\bar{y}z &= (y+\bar{y})(xz) \\ &= 1\cdot(xz) \\ &= xz \end{aligned}$$

这样，xz 也是一个表示这个电路的布尔表达式，但包含更少的算子。图 9-11 说明了这个电路的两个不同实现。第二个电路只使用一个门，但第一个却使用了三个门和一个反相器。

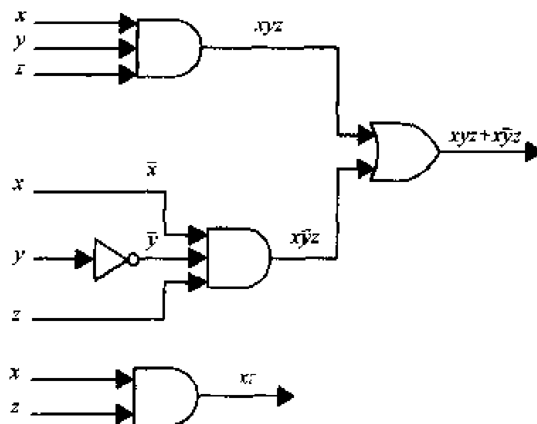


图 9-11 具有相同输出的两个电路

这个例子说明，在一个电路的积之和表达式中，将一些项合并会导出这个电路的更简单的表达式。下面将描述化简积之和展开式的两个过程，这两个过程的目的都是为了产生布尔积的布尔和，使其所包含的文字之积的个数最少，从而使得这些积中所包含的文字个数最少。

虽然现代电路常常是由比与门、或门和反相器更加复杂类型的元件构成，但本节所描述的化简积之和展开式的技术仍然有特殊的价值。虽然有各种各样的过程用来化简由这些更加复杂的元件构成的电路，但其中的许多方法所使用的思想与本节所描述的方法所使用的思想类似。

9.4.2 卡诺图

对于表示电路的一个布尔表达式，为了减少其中项的个数，有必要去发现可以合并的项。如果布尔函数所包含的变元相对较少，可以用一种图形法来发现能被合并的项。此法称为卡诺图，它是由摩里斯·卡诺（Maurice Karnaugh）[⊖]在 1953 年发现。他的方法是建立在更早的维奇（E.W.Veitch）的工作基础上的（维奇的方法通常只适用于六个以下变元的函数）。卡诺图给出了一种化简积之和展开式的可视化方法，但此法不适用于将化简过程机械化。下面说明怎么用卡诺图来化简包含两个变元的布尔函数的展开式。

在具有两个变元 x 和 y 的布尔函数的积之和展开式中，有四种可能的小项。具有这两个变元的布尔函数的卡诺图由四个方格组成，如果一个小项在此展开式中出现，则表示这个小项的方格就被放置 1。如果一些方格所表示的小项只在一个变元处不一样，则称这些方格是相邻的。例如，表示$\bar{x}y$ 的方格与表示 xy 的方格及表示$\bar{x}\,\bar{y}$ 的方格都相邻。四个方格及其表示的项如图 9-12 所示。

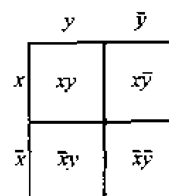


图 9-12 两个变元的卡诺图

例 1 找出下列各式的卡诺图。

a) $xy+\bar{x}y$ b) $x\bar{y}+\bar{x}y$ c) $x\bar{y}+\bar{x}y+\bar{x}\bar{y}$

⊖ 摩里斯·卡诺（Maurice Karnaugh）1924 年生于美国纽约市。他于纽约的城市大学获得学士学位，并从耶鲁大学获得硕士学位和博士学位。1952～1966 年期间，他担任贝尔实验室的一名技术人员，并在 1966～1970 年期间主管 AT&T 公司联邦系统分部的研究与开发部门。1970 年，他成为 IBM 的研究人员。卡诺为计算与远程通信的数字技术应用领域作出了基础贡献。目前，他的研究兴趣包括计算机中基于知识的系统和启发式探索方法。

解 当一个方格所表示的小项在积之和展开式中出现时，我们就在这个方格中放置一个 1。三个卡诺图如图 9-13 所示。 ■

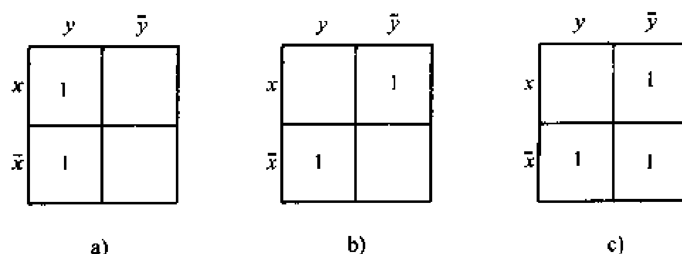


图 9-13 例 1 中积之和展开式的卡诺图

我们可以从卡诺图中识别出能够合并的小项。在卡诺图中，一旦有两个方格是相邻的，则由这两个方格所表示的小项就可被合并成一个积，且此积只涉及其中的一个变元。例如，$x\bar{y}$和$\bar{x}\bar{y}$是由两个相邻的方格表示的，它们可以合并成$\bar{y}$，因为$x\bar{y}+\bar{x}\bar{y}=(x+\bar{x})\bar{y}=\bar{y}$。而且，如果所有四个方格都是 1，则四个小项可以合并成一个项，即布尔表达式 1，它不涉及任何变元。在卡诺图中，如果一些小项能够合并，则我们将表示这些小项的方格所组成的块用圆圈圈起来，然后找出对应的积之和。其目的是找出可能最大的块，以及以最少的块来覆盖所有的 1，在此过程中，首先使用最大的块，并总是使用可能最大的块。

例 2 化简例 1 中的积之和展开式。

解 用这些展开式的卡诺图对小项进行分组的方式如图 9-14 所示。这些积之和式的极小展开式是 a) y、b) $x\bar{y}+\bar{x}y$ 和 c) $\bar{x}+\bar{y}$。 ■

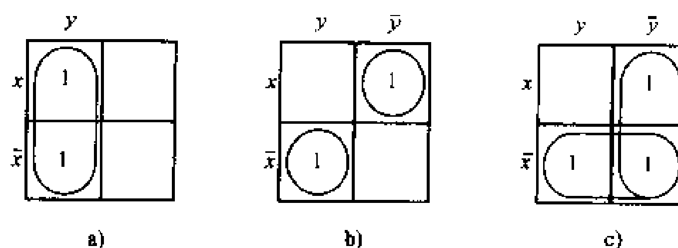


图 9-14 例 1 中积之和展开式的化简

三个变元的卡诺图是被分成八个方格的矩形，这些方格代表由三个变元组成的八个可能的小项。如果两个方格表示的小项只在一个文字处不一样，则它们称为是相邻的。一种画三个变元卡诺图的方法如图 9-15 a) 所示。这个卡诺图可以被认为是贴在圆柱体的表面上，如图 9-15 b) 所示。在这个圆柱体的表面上，两个方格有公共边界当且仅当它们是相邻的。

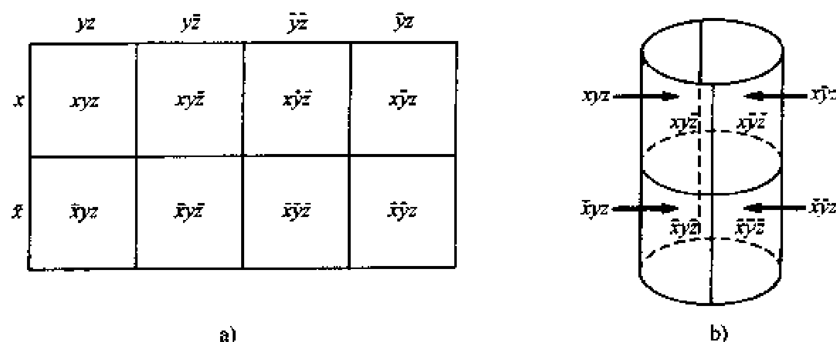


图 9-15 三个变元的卡诺图

为了化简三个变元的积之和展开式，我们用卡诺图来识别由可以合并的小项组成的块。两个相邻方格组成的块代表了一对小项，它们可以合并成两个文字的积，2×2 和 4×1 方格组成的块代表可以合并成一个文字的小项，全部八个方格组成的块代表函数 1，它不是任何文字的积。1×2、2×1、2×2、4×1 和 4×2 块及其代表的积如图 9-16 所示。

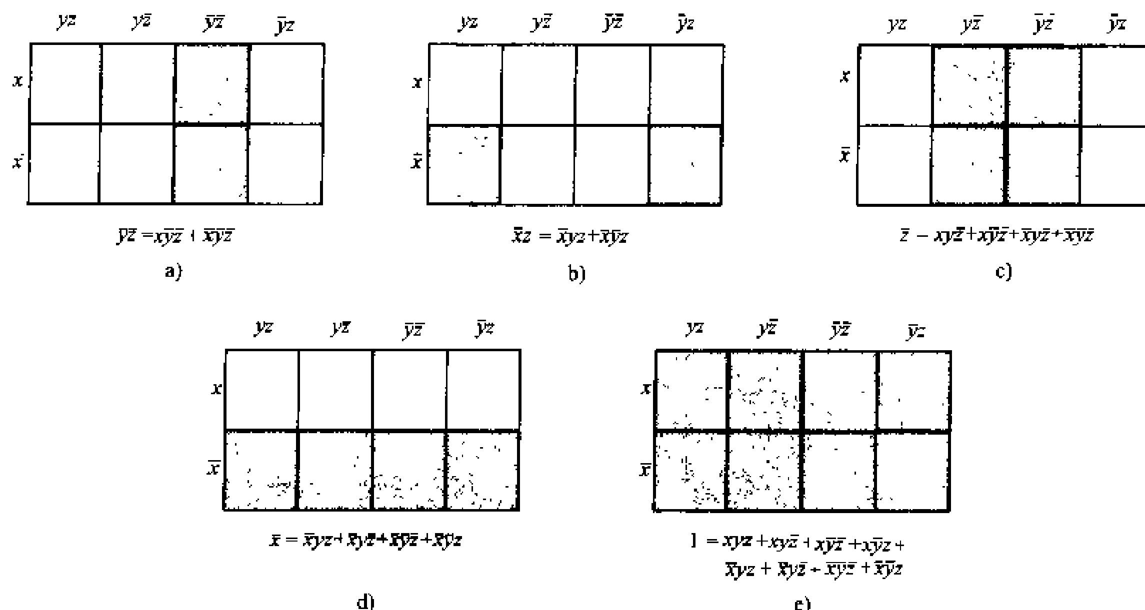


图 9-16 三个变元卡诺图中的块

例 3 用卡诺图化简下列积之和展开式。

a) $xy\bar{z}+x\bar{y}\bar{z}+\bar{x}yz+\bar{x}\bar{y}\bar{z}$ b) $x\bar{y}z+x\bar{y}\bar{z}+\bar{x}yz+\bar{x}\bar{y}z+\bar{x}\bar{y}\bar{z}$

c) $xyz+xy\bar{z}+x\bar{y}z+x\bar{y}\bar{z}+\bar{x}yz+\bar{x}\bar{y}z+\bar{x}\bar{y}\bar{z}$

解 这些积之和展开式的卡诺图如图 9-17 所示。块的分组表明：项数最少的布尔积之布尔和展开式为 a) $x\bar{z}+\bar{y}\bar{z}+\bar{x}yz$、b) $\bar{y}+\bar{x}z$ 和 c) $x+\bar{y}+z$。 ■

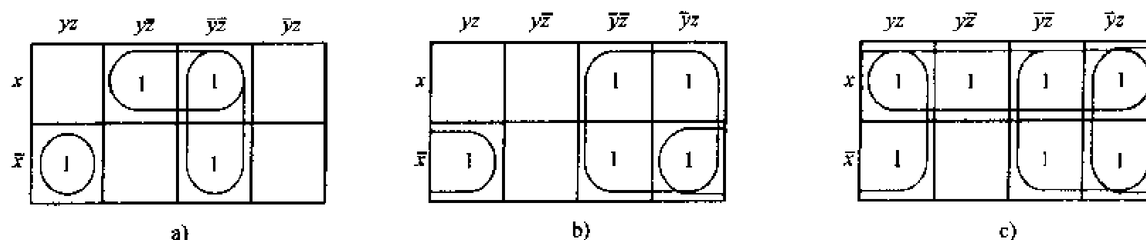


图 9-17 三个变元卡诺图的使用

四个变元的卡诺图是被分成 16 个方格的正方形，这些方格代表由四个变元组成的 16 个可能的小项。一种画四个变元卡诺图的方法如图 9-18 所示。

两个方格是相邻的当且仅当它们表示的小项只在一个文字处不一样。因而，每个方格都和另外四个方格相邻。四个变元的积之和展开式的卡诺图可以被认为是贴在圆环面上，因而相邻的方格具有公共的边界（见练习 20）。四个变元的积之和展开式的化简也是通过识别一

些块来实现的，这些块可能由 2、4、8 或 16 个方格组成，它们代表的小项可以合并。且每个表示小项的方格都必须被用来产生更少数量的文字的积，或者包含在展开式中。在图 9-19 中，我们给出了一些块，这些块表示三个文字的积、两个文字的积或单个文字。

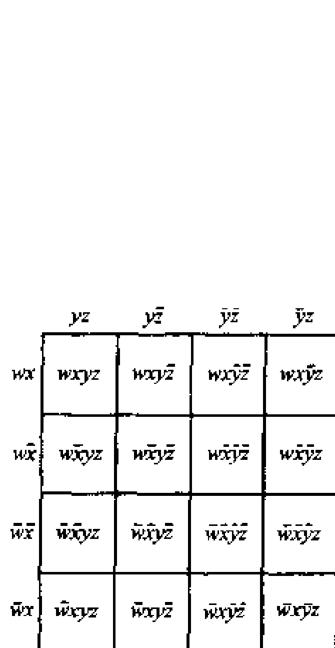


图 9-18 四个变元的卡诺图

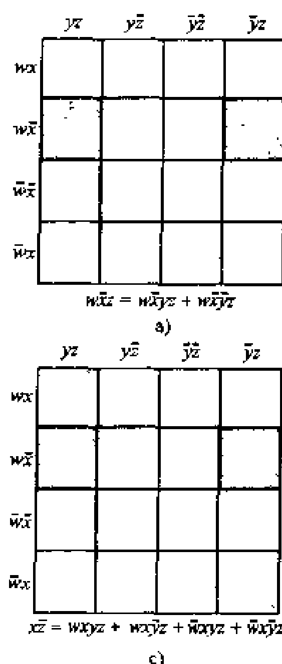
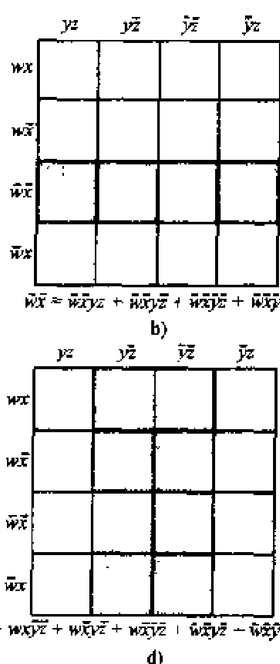
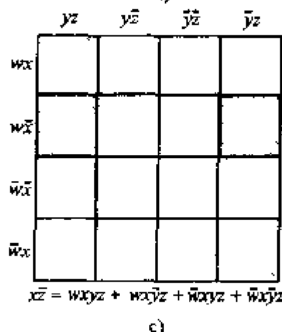
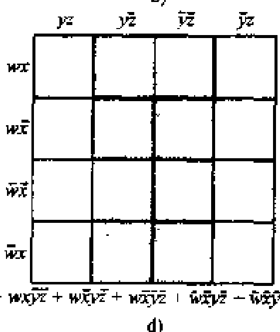





图 9-19 四个变元卡诺图中的块

就像两个或三个变元的卡诺图的作用一样，我们的目的也是在图中标出 1 构成的最大块，然后用最大块优先法则以最少的块覆盖所有的 1，我们也总是使用可能最大的块。

例 4 用卡诺图化简下列积之和展开式。

a) $wxyz + wxy\bar{z} + wx\bar{y}\,\bar{z} + w\bar{x}yz + w\bar{x}\,\bar{y}z + w\bar{x}\,\bar{y}\,\bar{z} + \bar{w}x\,\bar{y}z + \bar{w}\,\bar{x}yz + \bar{w}\,\bar{x}y\,\bar{z}$

b) $wx\,\bar{y}\,\bar{z} + w\bar{x}yz + w\bar{x}y\,\bar{z} + w\bar{x}\,\bar{y}\,\bar{z} + \bar{w}x\,\bar{y}\,\bar{z} + \bar{w}\,\bar{x}\,\bar{y}\,\bar{z}$

c) $wxy\,\bar{z} + wx\,\bar{y}\,\bar{z} + w\bar{x}yz + w\bar{x}y\,\bar{z} + w\bar{x}\,\bar{y}\,\bar{z} + \bar{w}xyz + \bar{w}xy\,\bar{z} + \bar{w}x\,\bar{y}\,\bar{z} + \bar{w}x\,\bar{y}z + \bar{w}\,\bar{x}y\,\bar{z} + \bar{w}\,\bar{x}\,\bar{y}\,\bar{z}$

解 这些展开式的卡诺图如图 9-20 所示。用所示的块可导出如下的积之和：a) $wyz + wx\,\bar{z} + w\,\bar{x}\,\bar{y} + \bar{w}\,\bar{x}y + \bar{w}x\,\bar{y}z$，b) $\bar{y}\,\bar{z} + w\bar{x}y + \bar{x}y\,\bar{z}$和 c) $\bar{z} + \bar{w}x + w\,\bar{x}y$。读者应该确定，在每一部分中是否可能选择其他的块，以产生表示这些布尔函数的不同的积之和。 ■

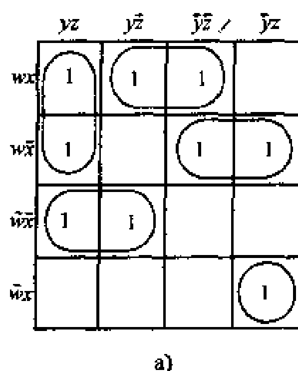
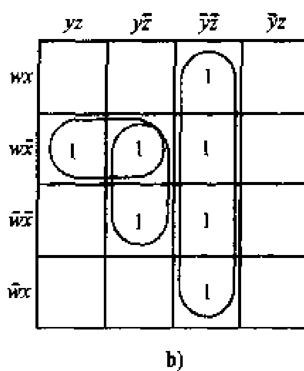
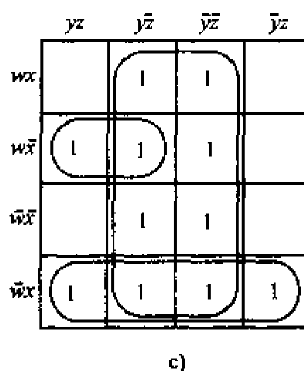




图 9-20 四个变元卡诺图的使用

9.4.3 无需在意条件

在某些电路中，由于输入值的一些组合从未出现过，所以我们只关心输入值的某些组合所产生的输出。这使得我们在生产具有所需输出的电路时有很大自由，因为对于所有不出现的输入值的组合，其输出值可以任意选择。函数对于这种组合的值称为*无需在意条件*。在卡诺图中，对于那些其函数值可以任意选择的变元值组合，用 d 对其作记号。在化简过程中，如果输入值的组合在卡诺图中导致最大的块，则我们可以将其赋值 1。下面的例子说明了这一点。

例 5 用二进制数字对十进制展开式进行编码的一种方法是：对十进制展开式中的每一位，用四个二进制数字对其编码。例如，873 的编码为 100001110011。十进制展开式的这种编码方式称为*十进制数二元编码*。因为有 16 个四位二进制数，但只有 10 个十进制数字，所以还有 6 个四位二进制数没有被用来对数位进行编码。假设现在需要构造一个电路使得：如果数位大于或等于 5，则输出 1；若数位小于 5，则输出 0。怎样仅用与门、或门和反相器来构造这个电路呢？

解 以 $F(w,x,y,z)$ 记此电路的输出，其中 $wxyz$ 是一个十进制数位的二进制表达式。F 的值如表 9-13 所示，图 9-21a 是 F 的卡诺图，其中的无需在意位置都是 d。我们可以将 d 包括在块中或者剔除出去，这样块就有很多可能的选择。例如，如果剔除所有的 d 方格，则形成块如图 9-21b 所示，所产生的表达式为 $w\bar{x}\bar{y}+\bar{w}xy+\bar{w}xz$。如果包括某些 d 而剔除其余的，则形成的块如图 9-21c 所示，且所产生的表达式为 $w\bar{x}+\bar{w}xy+x\bar{y}z$。最后，如果包括所有的块，且使用如图 9-21d 所示的块，则产生最简单的展开式，即 $F(w,x,y,z)=w+xy+xz$。 ■

表 9-13

数字	w	x	y	z	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1

9.4.4 奎因－莫可拉斯基方法

我们已经看到，可以用卡诺图将布尔函数展开为形如布尔积之布尔和的极小表达式。但当变元超过四个时，卡诺图就变得难以使用，而且卡诺图的使用还要依赖于用目测

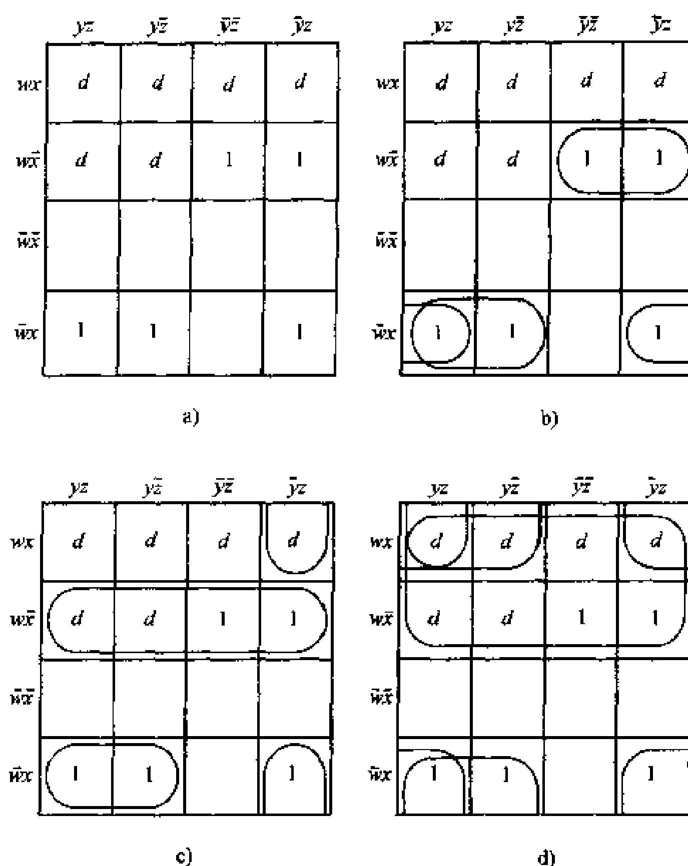


图 9-21　表明其无需在意条件位置的卡诺图

方法将项分成组。鉴于这些原因，需要可以机械化的过程来化简积之和展开式，奎因-莫可拉斯基方法就是这样一种过程，它可以用于含有任意多个变元的布尔函数。此法是由 W. V. 奎因和 E. J. 莫可拉斯基[○]于 20 世纪 50 年代提出。它基本上由两部分组成，第一部分寻找可

○ 爱德华·莫可拉斯基（Edward J. McCluskey）生于 1929 年，就读于 Bowdoin 学院和麻省理工学院，并于 1956 年在麻省理工学院获得电子工程学博士学位。他于 1955 年进入贝尔电话实验室，在那里一直待到 1959 年。从 1959 年到 1966 年，莫可拉斯基是普林斯顿大学的电子工程学教授，并在 1961 到 1966 年间在普林斯顿担任计算中心主任。他于 1967 年在斯坦福大学担任计算机科学和电子工程学的教授，并于 1969 到 1978 年间，在那里担任数字系统实验室的主任。莫可拉斯基对计算机科学的许多领域有过研究，包括容错计算、计算机体系结构、测试和逻辑设计。他现在是斯坦福大学的可靠性计算中心的主任，还是 ACM（美国计算机协会）的会员。

威纳德·奎因（Willard Van Orman Quine）1908 年出生于俄亥俄的阿克伦（Akron），先后就读于 Oberlin 学院和哈佛大学，于 1932 年在哈佛大学获哲学博士学位，并于 1933 年成为哈佛的年轻教员。由于他的才能，于 1936 年在哈佛获得任命。在他的职业生涯中，除了第二次世界大战外，他一直在哈佛工作。而在第二次世界大战期间，他服务于美国海军，任务是破译来自于德国潜艇的密码。奎因一直对算法感兴趣，而对硬件不感兴趣。他因发现现在称为奎因－莫可拉斯基的方法而成名，这是一个用于数理逻辑教学的装置，而不是用来化简开关电路。奎因是世界上最著名的哲学家之一，他对于知识理论、数理逻辑和集合论以及逻辑和语言的哲学都有着根本性的贡献。他的著作，例如发表于 1937 年的《*New Foundation of Mathematical Logic*》和发表于 1960 年的《*Word and Object*》都有着深远的影响。奎因于 1978 年从哈佛退休，但还继续来往于座落在 Beacon 山的家与办公室之间。他现在仍然使用着产于 1927 年的 Remington 牌打字机，他曾在这台打印机上打印过自己的博士论文。很久以前，他曾请人修理这台打印机，以便增加一些特殊的符号，并去掉第二个句号、第二个逗号和问号。当他被问到为何要去掉问号时，他答到：“你知道，我只管确定的事”。甚至在黑客词典中都有一个词叫“*quine*”，它指的是这样一种程序，在完成输出后，它复制自己的源代码。以给定的程序设计语言来构造最短的 *quine* 是黑客中流行的一个难题。

能包含在形如布尔积之布尔和的极小展开式中的候选项，第二部分才确定哪些项将真正使用。下面用例子来说明这个过程是怎样进行的。

例 6　下面说明怎么用奎因-莫可拉斯基方法寻找等价于 $xyz + x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$ 的极小展开式。

解　我们用一串二进制数位来表示此展开式中的小项。如果 x 出现，则第一位为 1；如果 $\bar{x}$ 出现，则第一位为 0。如果 y 出现，则第二位为 1；如果 $\bar{y}$ 出现，则第二位为 0。如果 z 出现，则第三位为 1；如果 $\bar{z}$ 出现，则第三位为 0。然后根据对应数位串中 1 的个数来对这些项进行分组。这些信息如表 9-14 所示。

表 9-14

小　项	数位串	1 的个数
xyz	111	3
$x\bar{y}z$	101	2
$\bar{x}yz$	011	2
$\bar{x}\bar{y}z$	001	1
$\bar{x}\bar{y}\bar{z}$	000	0

可以合并的小项只在一个文字处不同，所以，对于两个可以合并的小项，在表示它们的数位串中，1 的个数仅相差 1。当两个小项被合并成一个积时，这个积只含有两个文字。两个文字的积可以如下表示：以短划线来记不出现的变元。例如，数位串 101 和 001 所表示的小项 $x\bar{y}z$ 和 $\bar{x}\bar{y}z$ 可以合并成 $\bar{y}z$，而 $\bar{y}z$ 可以用串 −01 表示。表 9-15 列出了所有可以合并的成对小项以及它们所产生的积。

表 9-15

			步骤 1			步骤 2		
	项	数位串		项	串		项	串
1	xyz	111	(1, 2)	xz	1−1	(1,2,3,4)	z	−−1
2	$x\bar{y}z$	101	(1, 3)	yz	−11			
3	$\bar{x}yz$	011	(2, 4)	$\bar{y}z$	−01			
4	$\bar{x}\bar{y}z$	001	(3, 4)	$\bar{x}z$	0−1			
5	$\bar{x}\bar{y}\bar{z}$	000	(4, 5)	$\bar{x}\bar{y}$	00−			

下一步，对于由两个文字构成的积，如果两个这样的积能够合并，则将之合并成一个文字。两个这样的积能够合并的条件是：它们所包含的文字是两个相同变元的文字，并且只有其中一个变元的文字不一致。就表示这些积的串来说，它们必须在相同位置有一个短划线，且在其余的两个位置中必须恰好有一个位置的内容不相同。我们可以将串 −11 和 −01 所表示的积 yz 和 $\bar{y}z$ 合并成 z，而 z 用串 −−1 表示。所有能够以这种方式合并的项如表 9-15 所示。

在表 9-15 中，我们还指出了哪些项可以用来形成更少文字的积，这些项不一定在极小展开式中。下一步是找出积的一个极小集合，使之可以用来表示此布尔函数。我们从那些还没有被用来形成具有更少文字的积着手。再下一步，我们构造表 9-16，合并原来项所形成的每一个候选积构成此表的行，原来的项构成列。如果积之和展开式中原来的项被用来形成这个候选积，则在相应的位置打上 X，此时称此候选项**覆盖**了原来的小项。我们需要至少一个积，它覆盖原来的每一个小项。因而，一旦此表的某一列只有一个 X，则此 X 所在的行所对应的积必定被使用。从表 9-16 可以看出，z 和 $\bar{x}\bar{y}$ 都是必需的。所以，最后的答案是 $z+\bar{x}\bar{y}$。■

就像例6所说明的那样，奎因－莫可拉斯基方法用下面一系列步骤来化简积之和展开式。

表 9-16

	xyz	$x\overline{y}z$	$\overline{x}yz$	$\overline{x}\overline{y}z$	$\overline{x}\overline{y}\overline{z}$
z	X	X	X	X	
$\overline{x}\overline{y}$				X	X

1）将由 n 个变元构成的每一个小项表示成一个长度为 n 的二进制数串，如果 x_i 出现，则此串的第 i 个位置为1；如果 $\overline{x}_i$ 出现，则此串的第 i 个位置为0。

2）根据串中1的个数将串分组。

3）确定所有这样 $n-1$ 个变元的积，它们可以取为此展开式中小项的布尔和。将能够合并的小项表示成二进制数串，且这些串只在一个位置不相同。将这些 $n-1$ 个变元的积用如下的串表示：如果 x_i 出现在此积中，则此串的第 i 个位置为1；如果 $\overline{x}_i$ 出现，则此位置为0；如果此积中没有涉及 x_i 的文字，则此位置为短划线。

4）确定所有这样 $n-2$ 个变元的积，它们可以取为在前一个步骤形成的 $n-1$ 个变元的积的布尔和。将能够合并的 $n-1$ 个变元的积表示成如下的串：在同一位置有一个短划线，且只在一个位置不相同。

5）只要可能，继续将布尔积合并成更少变元的积。

6）找到所有这样的布尔积：它们虽然出现，但还没有被用来形成少一个文字的布尔积。

7）找到这些布尔积的最小集合，使得这些积的和表示此布尔函数。这可以用如下方法来完成：构造用一个表，列出哪些积覆盖了哪些小项。每一个小项必定被至少一个积覆盖。（这是此过程中最困难的部分，可用回溯法将其机械化。）

下面最后的例子说明了怎样用这个过程来化简四个变元的积之和展开式。

例7 用奎因－莫可拉斯基法化简积之和展开式

$$wxy\overline{z} + w\overline{x}yz + w\overline{x}y\overline{z} + \overline{w}xyz + \overline{w}x\overline{y}z + \overline{w}\overline{x}yz + \overline{w}\overline{x}\overline{y}z$$

解 首先将小项表示成二进制数串，然后根据串中1的个数来对项进行分组，如表9-17所示。表9-18给出了所有的布尔积，它们可以取为这些积的布尔和。

表 9-17

项	数位串	1的个数
$wxy\overline{z}$	1110	3
$w\overline{x}yz$	1011	3
$\overline{w}xyz$	0111	3
$w\overline{x}y\overline{z}$	1010	2
$\overline{w}x\overline{y}z$	0101	2
$\overline{w}\overline{x}yz$	0011	2
$\overline{w}\overline{x}\overline{y}z$	0001	1

表 9-18

			步骤1			步骤2		
	项	数位串		项	串		项	串
1	$wxy\overline{z}$	1110	(1, 4)	$wy\overline{z}$	1-10	(3,5,6,7)	$\overline{w}z$	0--1
2	$w\overline{x}yz$	1011	(2, 4)	$w\overline{x}y$	101-			
3	$\overline{w}xyz$	0111	(2, 6)	$\overline{x}yz$	-011			
4	$w\overline{x}y\overline{z}$	1010	(3, 5)	$\overline{w}xz$	01-1			
5	$\overline{w}x\overline{y}z$	0101	(3, 6)	$\overline{w}yz$	0-11			
6	$\overline{w}\overline{x}yz$	0011	(5, 7)	$\overline{w}\overline{y}z$	0-01			
7	$\overline{w}\overline{x}\overline{y}z$	0001	(6, 7)	$\overline{w}\overline{x}z$	00-1			

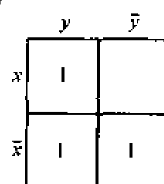
没有被用来形成更少变元之积的只有 $\overline{w}z$、$wy\overline{z}$、$w\overline{x}y$ 和 $\overline{x}yz$。表9-19表明了被每个这样的积覆盖的小项。为覆盖这些小项，必须包括 $\overline{w}z$ 和 $wy\overline{z}$，因为它们是分别覆盖 $\overline{w}xyz$ 和 $wxy\overline{z}$ 的唯一的积。一旦这两个积被包括进来，我们就可以看到，只有其中的一个是必要的。因而，$\overline{w}z + wy\overline{z} + w\overline{x}y$ 或者 $\overline{w}z + wy\overline{z} + \overline{x}yz$ 都可以看作是最后答案。■

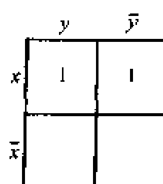
表 9-19

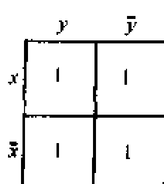
	$wxy\bar{z}$	$w\bar{x}yz$	$\bar{w}xyz$	$w\bar{x}y\bar{z}$	$\bar{w}x\bar{y}z$	$\bar{w}\bar{x}yz$	$\bar{w}\bar{x}\bar{y}z$
$\bar{w}z$			X		X	X	X
$wy\bar{z}$	X			X			
$w\bar{x}y$		X		X			
$\bar{x}yz$		X				X	

练习

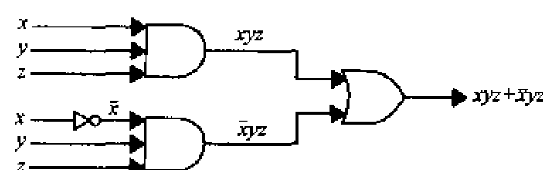
1. a) 画出二变元函数的卡诺图，并在表示$\bar{x}y$ 的方格中放置 1。
   b) 与上述方格相邻的方格所表示的小项是什么？
2. 寻找下列每个卡诺图所表示的积之和展开式。

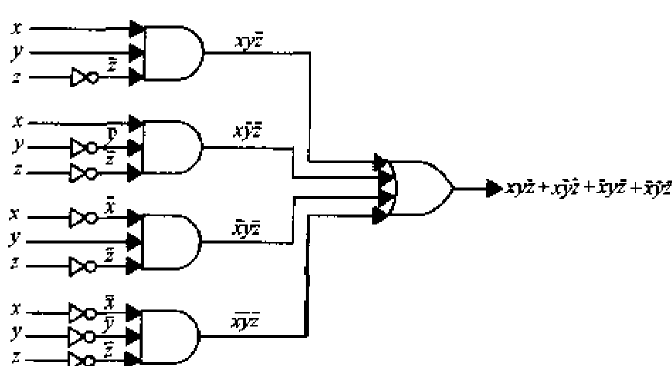
a)


b)


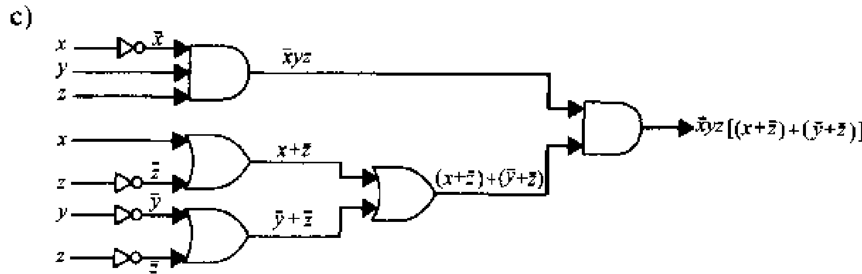
c)


3. 画出下列两个变元的积之和展开式的卡诺图。
   a) $x\bar{y}$　b) $xy+\bar{x}\bar{y}$　c) $xy+x\bar{y}+\bar{x}y+\bar{x}\bar{y}$
4. 用卡诺图找出下列关于变元 x 和 y 的布尔函数的极小展开式，且此展开式具有布尔积之布尔和的形式。
   a) $\bar{x}y+\bar{x}\bar{y}$　b) $xy+x\bar{y}$　c) $xy+x\bar{y}+\bar{x}y+\bar{x}\bar{y}$
5. a) 画出三变元函数的卡诺图，并在表示$\bar{x}y\bar{z}$的方格里放置 1。
   b) 与上述方格相邻的方格所表示的小项是什么？
6. 对于下列电路图，用卡诺图画出具有相同输出的更简单的电路图。

a)


b)


c)



7. 画出下列三变元积之和展开式的卡诺图。

   a) $x\bar{y}\bar{z}$　　　b) $\bar{x}yz+\bar{x}\bar{y}\bar{z}$　　　c) $xyz+xy\bar{z}+\bar{x}y\bar{z}+\bar{x}\bar{y}z$

8. 用卡诺图找出下列关于变元 x、y 和 z 函数的极小展开式，且此展开式具有布尔积之布尔和的形式。

   a) $\bar{x}yz+\bar{x}\bar{y}z$

   b) $xyz+xy\bar{z}+\bar{x}yz+\bar{x}y\bar{z}$

   c) $xy\bar{z}+x\bar{y}z+x\bar{y}\bar{z}+\bar{x}yz+\bar{x}\bar{y}z$

   d) $xyz+x\bar{y}z+x\bar{y}\bar{z}+\bar{x}yz+\bar{x}y\bar{z}+\bar{x}\bar{y}\bar{z}$

9. a) 画出四变元函数的卡诺图，并在表示$\bar{w}xy\bar{z}$的方格里放置 1。
   b) 与上述方格相邻的方格所表示的小项是什么？

10. 用卡诺图找出下列关于变元 w、x、y 和 z 函数的极小展开式，且此展开式具有布尔积之布尔和的形式。

    a) $wxyz+wx\bar{y}z+wx\bar{y}\bar{z}+w\bar{x}y\bar{z}+w\bar{x}\bar{y}z$

    b) $wxy\bar{z}+wx\bar{y}z+w\bar{x}yz+\bar{w}x\bar{y}z+\bar{w}\bar{x}y\bar{z}+\bar{w}\bar{x}\bar{y}z$

    c) $wxyz+wxy\bar{z}+wx\bar{y}z+w\bar{x}\bar{y}z+w\bar{x}\bar{y}\bar{z}+\bar{w}x\bar{y}z+\bar{w}\bar{x}y\bar{z}+\bar{w}\bar{x}\bar{y}z$

    d) $wxyz+wxy\bar{z}+wx\bar{y}z+w\bar{x}yz+w\bar{x}y\bar{z}+\bar{w}xyz+\bar{w}\bar{x}yz+\bar{w}\bar{x}y\bar{z}+\bar{w}\bar{x}\bar{y}z$

11. a) 五变元函数的卡诺图具有多少个方格？
    b) 在五变元函数的卡诺图中，对于任意给定的一个方格，有多少个方格与之相邻？

*12. 用卡诺图找出下列函数的极小展开式，使得此展开式具有布尔积之布尔和的形式，这些函数满足：其输入为十进制数字的二进制编码，其输出为 1，当且仅当对应于输入的数为
    a) 奇数　　　b) 不可由 3 整除　　　c) 不是 4、5 或 6

*13. 假设一个委员会中有五个成员，其中史密斯和琼斯的投票总与马库斯的投票相反。试用这个投票关系设计一个电路，实现此委员会的多数表决。

14. 使用奎因-莫可拉斯基法化简例 3 中的积之和展开式。

15. 使用奎因-莫可拉斯基法化简练习 8 中的积之和展开式。

16. 使用奎因-莫可拉斯基法化简例 4 中的积之和展开式。

17. 使用奎因-莫可拉斯基法化简练习 10 中的积之和展开式。

*18. 试解释怎么用卡诺图方法简化三个变元的和之积展开式。[提示：用 0 来标记展开式的极大项，然后构造极大项的块。]

19. 用练习 18 的方法化简和之积展开式$(x+y+z)(x+y+\bar{z})(x+\bar{y}+\bar{z})(\bar{x}+y+z)$。

*20. 在圆环面上画出四个变元的 16 个小项的卡诺图。

21. 用或门、与门和反相器构造一个电路，使得：当输入的十进制数字可以被 3 整除时输出 1，否则输出 0。其中，输入的十进制数字是二进制编码的十进制展开式。

对于练习 22～24，在所给的卡诺图中，d 表示无需在意条件。试找出它们的极小积之和展开式。

22.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	d	1	d	1
$w\bar{x}$		d	d	
$\bar{w}\bar{x}$		d	1	
$\bar{w}x$		1	d	

23.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			1
$w\bar{x}$		d	1	
$\bar{w}\bar{x}$		1	d	
$\bar{w}x$	d			d

24.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx		d	d	1
$w\bar{x}$	d	d	1	d
$\bar{w}\bar{x}$				
$\bar{w}x$	1	1	1	d

关键术语和结果

术语

布尔变元：只取 0 或 1 值的变元

$\bar{x}$（x 的补）：一个表达式，当 x 取值 0 时，它取值 1；当 x 取值 1 时，它取值 0

$x\cdot y$（或 xy）（x 与 y 的布尔积或合取）：一个表达式，当 x 和 y 都取值 1 时，它取值 1；否则取值 0

$x+y$（x 与 y 的布尔和或析取）：一个表达式，当 x 或 y 取值 1 时，或者当 x 和 y 都取值 1 时，它取值 1；否则取值 0

布尔表达式：如下递归得到的表达式：0，1，x_1，…，x_n 是布尔表达式；且如果 E_1 和 E_2 是布尔表达式，则$\overline{E_1}$、(E_1+E_2) 和 (E_1E_2) 也是布尔表达式

布尔表达式的对偶：通过交换 + 和·运算、0 和 1 得到的表达式

n 度布尔函数：从 B^n 到 B 的函数，其中$B=\{0,1\}$

布尔代数：具有两个二元运算 ∨ 和 ∧、元素 0 和 1、一元补运算 ¯ 的集合，它满足同一律、支配律、结合律、交换律和分配律

布尔变元的文字 x：或者为 x，或者为$\bar{x}$

x_1，x_2，…，x_n 的小项：布尔积 $y_1y_2\cdots y_n$，其中每个 y_i 或为x_i 或为$\bar{x}_i$

积之和展开式（或析取范式）：形如小项之析取的布尔函数的表示

函数完备：布尔运算符的一个集合被称为是函数完备的，如果每个布尔函数都能由这些布尔运算符表示

$x\mid y$（或 xNANDy）：一个表达式，当 x 和 y 都取值 1 时，它取值 0；否则取值 1

$x\downarrow y$（或 xNORy）：一个表达式，当 x 或 y 取值 1 时，或 x 和 y 都取值 1 时，它取值 0；否则取值 1

反相器：一种装置，它以布尔变元的值作为输入，产生输入的补

或门：一种装置，它以两个或更多布尔变元的值作为输入，输出它们的布尔和

与门：一种装置，它以两个或更多布尔变元的值作为输入，输出它们的布尔积

半加器：一种电路，它将两个二进制数字相加，产生一个和位与一个进位

全加器：一种电路，它将两个二进制数字及一个进位相加，产生一个和位与一个进位

***n* 个变元的卡诺图**：被分成 2^n 多个方格的矩形，每个方格表示这些变元的一个小项

结果

布尔代数中的恒等式（见第 9.1 节的表 9-5）。

对于布尔表达式表示的布尔函数间的任意恒等式，如将恒等式的两边取对偶，则等式依然成立。

每个布尔函数都可由积之和展开式表示。

集合 $\{+, ^{-}\}$ 和 $\{\cdot, ^{-}\}$ 都是函数完备的。

集合 $\{\downarrow\}$ 和 $\{|\}$ 都是函数完备的。

使用卡诺图来极小化布尔表达式。

使用奎因－莫可拉斯基来极小化布尔表达式。

复习题

1. 给出 n 度布尔函数的定义。
2. 有多少个 2 度布尔函数？
3. 给出布尔表达式集合的递归定义。
4. a) 什么是布尔表达式的对偶？
   b) 什么是对偶原理？怎么应用它找到关于布尔表达式的新的恒等式？
5. 试解释怎么构造一个布尔函数的积之和展开式。
6. a)“由运算符构成的一个集合是函数完备的”是什么含义？
   b) 集合 $\{+, \cdot\}$ 是函数完备的吗？
   c) 有没有单运算符构成的集合是函数完备的？
7. 试解释怎样用或门、与门和反相器构造一个电路，它用两个开关控制一个灯。
8. 用或门、与门和反相器构造一个半加器。
9. 是否有这样一种逻辑门，利用它可以构造或门、与门和反相器所能构造的所有电路？
10. a) 解释怎样用卡诺图来化简三个变元的积之和展开式。
    b) 用卡诺图化简积之和展开式 $xyz + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z}$。
11. a) 解释怎样用卡诺图来化简四个布尔变元的积之和展开式。
    b) 用卡诺图化简积之和展开式：
    $$wxyz + wxy\bar{z} + wx\bar{y}z + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}z + \bar{w}xyz + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$$
12. a) 什么是无需在意条件？
    b) 试解释怎么用无需在意条件由或门、与门和反相器构造这样一个电路：当十进制数字大于等于 6 时输出 1；当这个数字小于 6 时输出 0。
13. a) 试解释怎样用奎因－莫可拉斯基方法来化简积之和展开式。
    b) 用这个方法化简 $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$。

补充练习

1. 布尔变元 x、y 和 z 如何取值，可使下列等式成立？

   a) $x+y+z=xyz$ b) $x(y+z)=x+yz$ c) $\bar{x}\,\bar{y}\,\bar{z}=x+y+z$

2. 设 x 和 y 属于 {0, 1}。如果存在 {0, 1} 中的值 z 使得下列各式成立，有无必要使 $x=y$ 成立？

   a) $xz=yz$ b) $x+z=y+z$ c) $x\oplus z=y\oplus z$

   d) $x\downarrow z=y\downarrow z$ e) $x\mid z=y\mid z$

   布尔函数 F 称为是自对偶的当且仅当 $F(x_1,\cdots,x_n)=\overline{F(\bar{x}_1,\cdots,\bar{x}_n)}$。

3. 下列函数哪些是自对偶的？

   a) $F(x,y)=x$ b) $F(x,y)=xy+\bar{x}\,\bar{y}$

   c) $F(x,y)=x+y$ d) $F(x,y)=xy+\bar{x}y$

4. 试给出一个三变元自对偶布尔函数的例子。

*5. 有多少个 n 度布尔函数是自对偶的？

在 n 度布尔函数构成的集合上，定义关系≤，使得：$F\le G$ 当且仅当若 $F(x_1,x_2,\cdots,x_n)=1$ 就有 $G(x_1,x_2,\cdots,x_n)=1$。

6. 对于下列函数对，确定是否有 $F\le G$ 或 $G\le F$。

   a) $F(x,y)=x$，$G(x,y)=x+y$

   b) $F(x,y)=x+y$，$G(x,y)=xy$

   c) $F(x,y)=\bar{x}$，$G(x,y)=x+y$

7. 设 F 和 G 是 n 度布尔函数，证明：

   a) $F\le F+G$ b) $FG\le F$

8. 设 F、G 和 H 都是 n 度布尔函数，证明：$F+G\le H$，当且仅当 $F\le H$ 且 $G\le H$。

*9. 证明≤关系是 n 度布尔函数集合上的一个偏序关系。

*10. 画出由 16 个 2 度布尔函数（如第 9.1 节表 9-3 所示）的集合构成的部分有序集在偏序≤下的哈斯图。

*11. 对于下列每个等式，或者证明其为恒等式，或者找到变元的一组值使之不成立。

   a) $x\mid(y\mid z)=(x\mid y)\mid z$

   b) $x\downarrow(y\downarrow z)=(x\downarrow y)\downarrow(x\downarrow z)$

   c) $x\downarrow(y\mid z)=(x\downarrow y)\mid(x\downarrow z)$

定义布尔运算符⊙如下：$1\odot1=1$，$1\odot0=0$，$0\odot1=0$，$0\odot0=1$。

12. 证明 $x\odot y=xy+\bar{x}\,\bar{y}$。

13. 证明 $x\odot y=\overline{(x\oplus y)}$。

14. 证明下列各恒等式成立。

   a) $x\odot x=1$ b) $x\odot\bar{x}=0$ c) $x\odot y=y\odot x$

15. $(x\odot y)\odot z=x\odot(y\odot z)$ 是否总成立？

*16. 确定集合 {⊙} 是不是函数完备的。

*17. 在 16 个两变元 x 和 y 组成布尔函数中，有多少个能够用下列运算符集、变元 x 和 y 以及值 0 和 1 表示？

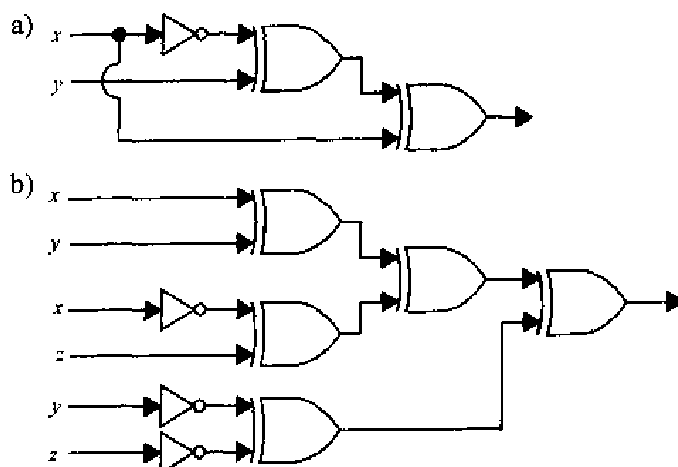
a) $\{^-\}$ b) $\{\cdot\}$ c) $\{+\}$ d) $\{\cdot, +\}$

异或门的记号如右，它从 x 和 y 产生输出 $x \oplus y$。

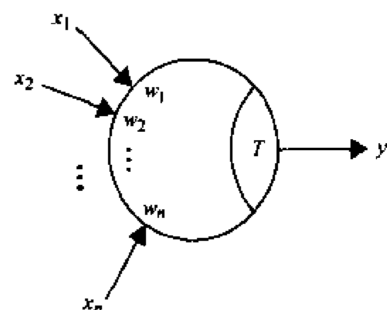


18. 确定下列电路 a) 和 b) 的输出。

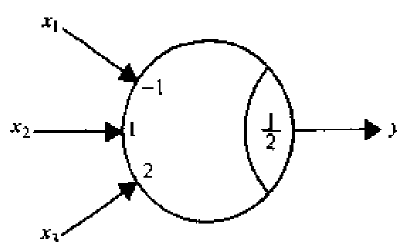
19. 如果除了或门、与门和反相器之外，还可以使用异或门，说明怎样用比第 9.3 节图 9-8 中所用的更少的门来构造一个半加器。



20. 试设计一个电路来确定：在一个四人委员会中，是否有三人或更多的人就某事投了赞成票，其中的每个人用一个开关来投票。



给定布尔变元 $x_1, x_2, \cdots, x_n$ 的一组输入值，阈值门产生输出 y，其中 y 为 0 或 1。每个阈值门都有一个阈值 T 以及一组权 $w_1, w_2, \cdots, w_n$，其中 T 及 $w_1, w_2, \cdots, w_n$ 都是实数。阈值门的输出 y 是 1 当且仅当 $w_1x_1 + w_2x_2 + \cdots + w_nx_n \geqslant T$。具有阈值 T 及权 $w_1, w_2, \cdots, w_n$ 的阈值门如右图所示。阈值门对于神经生理学和人工智能的建模都非常有用。



21. 阈值门表示了一个布尔函数。试找出由下面阈值门表示的布尔函数的布尔表达式。

22. 能够由阈值门表示的布尔函数称为阈值函数。证明下列每个函数都是阈值函数。

a) $F(x) = \overline{x}$ b) $F(x, y) = x + y$ c) $F(x, y) = xy$

d) $F(x, y) = x \mid y$ e) $F(x, y) = x \downarrow y$ f) $F(x, y, z) = x + yz$

g) $F(w, x, y, z) = w + xy + z$ h) $F(w, x, y, z) = wxz + x\overline{y}z$

*23. 证明 $F(x, y) = x \oplus y$ 不是阈值函数。

*24. 证明 $F(w,x,y,z)=wx+yz$ 不是阈值函数。

计算机题目

写出具有下列输入和输出的程序。

1. 给定两个布尔变元 x 和 y 的值，计算 $x+y$、xy、$x\oplus y$、$x|y$ 和 $x\downarrow y$ 的值。
2. 构造一个表，列出所有 256 个 3 度布尔函数的值。
3. 给定一个 n 元布尔函数的所有值，其中 n 是个正整数，构造这个函数的积之和展开式。
4. 给定一个布尔函数值的列表，只用运算符·和¯表示这个函数。
5. 给定一个布尔函数值的列表，只用运算符+和¯表示这个函数。

*6. 给定一个布尔函数值的列表，只用运算符|表示这个函数。

*7. 给定一个布尔函数值的列表，只用运算符↓表示这个函数。

8. 给定一个 3 度布尔函数值的列表，构造它的卡诺图。
9. 给定一个 4 度布尔函数值的列表，构造它的卡诺图。

**10. 给定一个布尔函数值的列表，用奎因–莫可拉斯基方法寻找这个函数的极小积之和表示。

11. 对于一个阈值门和 n 个布尔变元的值，给定它的阈值和一组权，确定这个门的输出。
12. 给定一个正整数，构造一个 n 元随机布尔表达式，且为析取范式。

计算和研究

用你已经写出的程序做下列练习。

1. 计算 7、8、9、10 度布尔函数的个数。
2. 构造 3 度布尔函数的表。
3. 构造 4 度布尔函数的表。
4. 将每个不同的三元布尔表达式表示成仅含与非运算符的析取范式，所使用的与非运算符越少越好。所需与非运算符的最大数量是多少？
5. 将每个不同布尔表达式表示成含有四个变元和仅含 NOR 运算符的析取范式，所使用的 NOR 运算符越少越好。所需 NOR 运算符的最大数量是多少？
6. 随机构造 10 个不同的四元布尔表达式，并确定使用奎因–莫可拉斯基方法极小化它们所需要的平均步骤数。
7. 随机构造 10 个不同的五元布尔表达式，并确定使用奎因–莫可拉斯基方法极小化它们所需要的平均步骤数。

写作题目

用课本以外的资料解决下列问题，并写成短文。

1. 描述一些早期设计的、用来解逻辑问题的机器，如 Stanhope 示范器、杰文（Javons）的逻辑机以及马昆德机器（Marquand Machine）。
2. 解释组合电路与顺序电路间的差别，然后解释怎样用触发器构造顺序电路。
3. 定义移位寄存器，且讨论怎样使用移位寄存器。说明怎样用触发器和逻辑门构造移位寄存器。

4. 说明怎样用逻辑门构造乘法器。
5. 找出逻辑门的物理构造。讨论在构造电路时，是否要用到与非门和或非门。
6. 解释怎样用相关性记号描述复杂的开关电路。
7. 描述怎样用乘法器构造开关电路。
8. 以用阈值门构造半加器和全加器为例，解释用阈值门构造开关电路的优点。
9. 描述无危险开关电路概念，并给出一些设计这样电路的原则。
10. 解释怎样用卡诺图将五元或六元函数极小化。
11. 描述 n 元布尔函数的函数分解的含义。讨论将布尔函数分解为元数更少的布尔函数的过程。

第 10 章 计算模型

计算机能够执行许多任务。每个任务都有两个问题：第一，它能否由计算机来完成？一旦知道这第一个问题的答案是肯定的，就会问第二个问题，怎么执行这个任务？计算模型就是用来帮助回答这些问题的。

下面将讨论三种类型的计算模型：文法、有限状态机和图灵机。文法是用来产生一种语言中的词，并且确定一个词是否属于一个语言。文法产生的形式语言不仅可以作为自然语言的模型，如英语，还可以作为程序设计语言的模型，如 Pascal、Fortran、Prolog 和 C。特别是，文法在编译器的理论和构造中极为重要。在 20 世纪 50 年代，美国语言学家诺姆·乔姆斯基（Noam Chomsky）首先使用了下面将要讨论的文法。

在建模中，使用着各种类型的有限状态机。每个有限状态机都有一个状态集合（包含初始状态）和一个输入字母表，还有一个转移函数，它对每个由状态和输入构成的对，指定下一个状态。有限状态机的状态使得它具有有限的存储能力。有些有限状态机对每个转移产生一个输出符号。这类机器可以用作许多种机器的模型，如自动售货机、延迟机、二进制数加法器和语言识别器。我们还将讨论没有输出但具有终结状态的有限状态机，这样的机器广泛用于语言的识别，它们所识别的串是这样的串：由初始状态运行到终结状态。文法和有限状态机的概念具有紧密的联系，我们将要刻画有限状态机所能识别的集合的特征，并且证明这些集合恰恰就是某种类型文法所产生的集合。

然后引入图灵机的概念。我们将说明怎么用图灵机来识别集合，还要说明怎么用图灵机来计算数论函数。最后讨论图灵－丘奇论题：每个能行的计算都可由图灵机来完成。

10.1 语言和文法

10.1.1 引言

英语中，单词能以各种方式进行组合，单词的哪些组合可以构成有效句子是由英语语法确定的。例如：the frog writes neatly（青蛙的字写得很整洁）是一个有效的句子，因为它是由一个名词短语 the frog 接一个动词短语 writes neatly 构成的，其中：名词短语 the frog 是由冠词 the 和名词 frog 组成的，动词短语 writes neatly 是由动词 writes 和副词 neatly 组成。我们并不在意这是一个毫无意义的句子，因为我们只关心句子的*语法*，或者说形式，而不在意它的*语义*，或者说含义。我们也要指出，词的组合 swims quickly mathematics 不是有效的句子，因为它不符合英语语法。

自然语言(即口头语言)，如英语、法语、德语或西班牙语，都极为复杂。事实上，对一个自然语言，看起来不大可能说出它的所有语法规则。将一个语言自动翻译成另一个语言的研究了引出了*形式语言*的概念。与自然语言不同，形式语言是由一组意义明确的语法规则定义的，语法规则不仅对于语言学和自然语言的研究十分重要，而且对于程序设计语言的研究也很重要。

形式语言的句子是用语法来描述的。在程序设计语言的应用中，经常出现两类问题：

(1) 怎么能够确定一组单词是否组合成了形式语言的一个有效句子?(2) 怎么才能产生形式语言的一个有效句子。在考虑这两类问题时，文法的使用十分有益。

在给出文法的技术定义之前，先描述文法的一个例子，这个例子产生英语的一个子集，此英语子集是用下列规则定义的，这些规则描述了怎么产生有效的句子。这些规则是：

1. 句子是由一个名词短语后接一个动词短语形成的；
2. 名词短语由一个冠词接一个形容词再接一个名词组成，或者
3. 名词短语由一个冠词接一个名词组成；
4. 动词短语由一个动词接一个副词组成，或者
5. 动词短语由一个动词组成；
6. 冠词是 *a*，或者
7. 冠词是 *the*；
8. 形容词是 *large*，或者
9. 形容词是 *hungry*；
10. 名词是 *rabbit*，或者
11. 名词是 *mathematician*；
12. 动词是 *eats*，或者
13. 动词是 *hops*；
14. 副词是 *quickly*，或者
15. 副词是 *wildly*。

从这些规则出发，使用一系列替代直到不能再应用规则，就能形成一个有效的句子。例如，沿着下列替代序列就能得到一个有效句子：

句子
名词短语　动词短语
冠词　形容词　名词　动词短语
冠词　形容词　名词　动词　副词
the 形容词　名词　动词　副词
the large 名词　动词　副词
the large rabbit 动词　副词
the large rabbit hops 副词
the large rabbit hops quickly

容易看出，下面都是有效句子：*a hungry mathematician eats wildly*，*a large mathematician hops*，*the rabbit eats quickly*，等等。也可以看出，*the quickly eats mathematican* 不是有效句子。

10.1.2　短语结构文法

在给出文法的形式定义之前，先引入一个小术语。

定义 1　词汇表（字母表）V 是一个有限的非空集合，其元素称为符号。V 上一个词（或句子）是由 V 中元素组成的有限长度的串。空串（或零串）是没有符号的串，记为 λ。V 上所有词的集合记为 V^*。V 上的一个语言是 V^* 的一个子集。

注意：空串 λ 是不包含任何符号的串。它不同于空集 $\emptyset$。因而 $\{\lambda\}$ 是仅包含一个串的集合，此串为空串。

可以用多种方式来指明语言。一种方式是列出语言中的所有词；还有一种方式是给出一些标准，使得一个词要在这个语言中，就必须满足这些标准。本节将描述另一种重要方式：使用文法，如使用本节引言中给出的规则集合。为产生词，文法提供一个由各种类型符号组成的集合和一个由规则组成的集合。更确切地说，文法有一个**词汇表** V，V 是一个由符号组成的集合，语言中的成分就是这些符号导出的。词汇表中的某些元素不能由其他符号替换，这些元素称为**终结符**；词汇表中的其他成分可以用其他符号替换，它们被称为**非终结符**。终结符和非终结符集合通常分别记为 T 和 N。在本节引言所给的例子中，终结符集是 $\{a, the, rabbit, mathematician, hops, eats, quickly, wildly\}$，非终结符集是 {句子，名词短语，动词短语，形容词，冠词，名词，动词，副词}。词汇表中有一个特殊的元素，我们总是从这个特殊元素开始定义其他符号，此符号称为**初始符**，记为 S。在引言的例子中，初始符是句子。由词汇表 V 中元素构成的所有串的集合记为 V^*。指明 V^* 中的串能被什么样的串替代的规则称为文法的**产生式**，指明 w_0 可以替换为 w_1 的产生式记为 $w_0 \to w_1$。在本节引言所给的文法中，我们列举了所有产生式。如果使用刚才定义的记号，其中第一个产生式为句子→名词短语　动词短语。我们以下列定义作为小结。

定义 2　一个**短语结构文法** $G=(V,T,S,P)$ 由下列四部分组成：词汇表 V，由 V 的所有终结符组成的 V 的子集合 T，V 的初始符 S，和产生式集合 P。集合 $V-T$ 记为 N，N 中的元素称为非终结符。P 中的每个产生式的左边必须至少包含一个非终结符。

例 1　设 $G=(V,T,S,P)$，其中：$V=\{a,b,A,B,S\}$，$T=\{a,b\}$，S 是初始符，$P=\{S\to ABa, A\to BB, B\to ab, AB\to b\}$。则 G 是一个短语结构文法的例子。■

我们对短语结构文法的产生式所产生的词感兴趣。

定义 3　设 $G=(V,T,S,P)$ 是一个短语结构文法，$w_0=lz_0r$（即 l，z_0 和 r 的连接）和 $w_1=lz_1r$ 是 V 上的串。若 $z_0\to z_1$ 是 G 的一个产生式，则称由 w_0 **可直接派生** w_1，记为 $w_0 \Rightarrow w_1$。如果 V 上的串 w_0，w_1，…，w_n（$n\geq 0$）满足 $w_0\Rightarrow w_1$，$w_1\Rightarrow w_2$，…，$w_{n-1}\Rightarrow w_n$，则称由 w_0 **可派生** w_n，记为 $w_0 \stackrel{*}{\Rightarrow} w_n$。由 w_0 得到 w_n 的序列称为**派生**。

例 2　在例 1 的文法中，由串 Aba 可直接派生 $Aaba$，因为 $B\to ab$ 是此文法中的一个产生式。由串 ABa 可派生 $abababa$，因为接连使用产生式 $B\to ab$，$A\to BB$，$B\to ab$ 和 $B\to ab$，可得 $ABa\Rightarrow Aaba\Rightarrow BBaba\Rightarrow Bababa\Rightarrow abababa$。■

定义 4　设 $G=(V,T,S,P)$ 是一个短语结构文法，由 G **生成的语言**（或 G 的语言）是初始符 S 能够派生的所有终结符串构成的集合，记为 $L(G)$。即：

$$L(G)=\{w\in T^* \mid S\stackrel{*}{\Rightarrow} w\}$$

下面两个例子都是寻找短语结构文法所生成的语言。

例 3　设 G 是一个文法，其词汇表为 $V=\{S,A,a,b\}$，终结符集 $T=\{a,b\}$，初始符为 S，产生式为 $P=\{S\to aA, S\to b, A\to aa\}$。求这个文法的语言 $L(G)$。

解　使用产生式 $S\to aA$，可以从初始符 S 派生 aA，还可用产生式 $S\to b$ 派生 b。使用

产生式 $A \to aa$，可以从 aA 派生 aaa。没有其他的词还能被派生，故 $L(G) = \{b, aaa\}$。■

例 4 设 G 是一个文法，其词汇表为 $V = \{S, 0, 1\}$，终结符集 $T = \{0, 1\}$，初始符为 S，产生式为 $P = \{S \to 11S, S \to 0\}$。求这个文法的语言 $L(G)$。

解 分别使用 $S \to 0$ 和 $S \to 11S$，可以从 S 派生出 0 和 $11S$。从 $11S$ 可以派生出 110 和 $1111S$。从 $1111S$ 可以派生出 11110 和 $111111S$。在派生过程的每一步，或者在串的末尾加两个 1，或者在串的末尾加 0 后终止派生。总之，$L(G) = \{0, 110, 11110, 111110, \cdots\}$，即 $L(G)$ 是如下串的集合：开始是偶数多个 1，最后是一个 0。这个结论可如下归纳证明：使用 n 次产生式之后，所生成的终结符串只能是这样的串：先是 $n-1$ 个或更少个 11 的连接，后面跟一个 0（留作练习）。■

经常出现的问题是要构造一个文法来生成一个给定的语言。下面的三个例子描述了这样问题。

例 5 给出生成集合 $\{0^n 1^n \mid n = 0, 1, 2, \cdots\}$ 的一个短语结构文法。

解 此集合中的元素是这样的串：先是一串 0，后跟同样多个 1。可以用两个产生式来生成所有这些串（包括空串），第一个产生式用来不断地产生此语言中更长的串，方法是在前面加一个 0，末尾加一个 1；第二个产生式以空串来替代 S。所求的文法是 $G = \{V, T, S, P\}$，其中 $V = \{0, 1, S\}$，终结符集 $T = \{0, 1\}$，初始符为 S，产生式为

$$S \to 0S1$$

$$S \to \lambda$$

此文法能够生成所给集合的证明作为练习留给读者。■

上面这个例子讨论的是如下串的集合：先是一串 0，后跟一串 1，其中 0 的个数和 1 的个数相同。下面的例子还是讨论这样的串，但 0 的个数与 1 的个数不一定相同。

例 6 给出生成集合 $\{0^m 1^n \mid m$ 和 n 和为非负整数$\}$ 的一个短语结构文法。

解 下面构造两个文法 G_1 和 G_2 来生成这个集合，这也说明了两个文法可能生成相同的语言。

文法 G_1 的字母表 $V = \{S, 0, 1\}$，终结符集 $T = \{0, 1\}$，产生式为 $S \to 0S$，$S \to S1$ 和 $S \to \lambda$。G_1 能生成所给集合是因为：应用第一个产生式 m 次就在串的前面增加了 m 个 0，应用第二个产生式 n 次就在串的后增加了 n 个 1。详细证明留作练习。

文法 G_2 的字母表 $V = \{S, V, 0, 1\}$，终结符集 $T = \{0, 1\}$，产生式为 $S \to 0S$，$S \to 1A$，$S \to 1$，$A \to 1A$，$A \to 1$ 和 $S \to \lambda$。G_2 也能生成所给集合的详细证明留作练习。■

有时候，一些很容易描述的集合不得不用非常复杂的文法来生成，下面就是一个这样的例子。

例 7 生成集合 $\{0^n 1^n 2^n \mid n = 0, 1, 2, 3, \cdots\}$ 的一个文法是：$G = \{V, T, S, P\}$，其中 $V = \{0, 1, 2, S, A, B\}$，终结符集 $T = \{0, 1, 2\}$，初始符为 S，产生式有 $S \to 0SAB$，$S \to \lambda$，$BA \to AB$，$0A \to 01$，$1A \to 11$，$1B \to 12$，$2B \to 22$。此命题的正确性证明留作练习。在本节后面指出的意义下，会明白此文法是生成这个语言的最简单类型的文法。读者也许奇怪这个文法是怎么得来的，因为似乎很难凭空想出这样一个文法。但如果知道可以用计算理论中的技术

系统地构造出这个文法，也许就不奇怪了，但这种技术已超出了本书的范围。■

10.1.3 短语结构文法的类型

短语结构文法可以根据其产生式的类型来分类。下面将来描述诺姆·乔姆斯基[○]引入的分类方法。在 10.4 节将会看到，以这种方法定义的不同语言类型，与不同的计算机器模型识别的语言类相对应。

0 型文法对其产生式没有限制。1 型文法只有两种形式的产生式：一种是 $w_1 \to w_2$ 形式的产生式，其中 w_2 的长度大于或等于 w_1 的长度；另一种是 $w_1 \to \lambda$。2 型文法只有形如 $w_1 \to w_2$ 的产生式，其中 w_1 是一个非终结符的单个符号。3 型文法的产生式 $w_1 \to w_2$ 必须满足 $w_1 = A$，且 $w_2 = aB$ 或 $w_2 = a$，其中 A 和 B 是非终结符，a 是终结符，或者满足 $w_1 = S, w_2 = \lambda$。

从定义可以看出，3 型文法都是 2 型文法，2 型文法都是 1 型文法，1 型文法都是 0 型文法。2 型文法又称为*上下文无关文法*，因为如果某个串中出现了一个非终结符，而此非终结符又是某个产生式的左边，则不管此串中的其他符号是什么，这个非终结符都可被换掉。2 型文法生成的语言称为*上下文无关语言*。当一个文法具有形如 $lw_1r \to lw_2r$（而不是形如 $w_1 \to w_2$）的产生式时，这样的文法称为 1 型文法或上下文相关文法，因为只有当 w_1 被串 l 和 r 包围时，才能被替换为 w_2。3 型文法又称为*正则文法*。正则文法生成的语言称为是*正则的*。10.4 节讨论正则语言和有限状态机之间的关系。图 10-1 的文氏图说明了这些不同类型文法间的关系。

例 8 由例 6 可知，$\{0^m1^n \mid m, n = 0,1,2,\cdots\}$ 是一个正则语言，因为它是由一个正则文法生成的，即由例 6 的文法 G_2 生成的。■

例 9 由例 5 可知，$\{0^n1^n \mid n = 0,1,2,\cdots\}$ 是一个上下文无关语言，因为这个文法的产生式为 $S \to 0S1$ 和 $S \to \lambda$。在 10.4 节中我们将证明它不是正则语言。■

例 10 集合 $\{0^n1^n2^n \mid n = 0,1,2,3,\cdots\}$ 是一个上下文相关语言，因为它是由例 7 中的 1 型文法生成的。但它不是 2 型语言（如本章补充练习中的练习 28 所证）。■

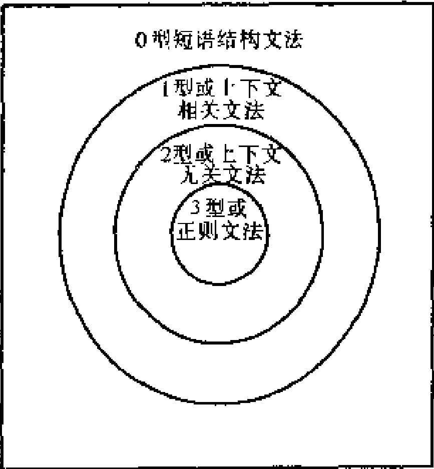


图 10-1 文法的类型

表 10-1 概括了用来对短语结构文法进行分类的术语。

表 10-1 文法的类型

类型	对产生式 $w_1 \to w_2$ 的限制	类型	对产生式 $w_1 \to w_2$ 的限制
0	无限制	2	$w_1 = A$ 其中 A 是非终结符
1	$l(w_1) \leq l(w_2)$，或 $w_2 = \lambda$	3	$w_1 = A$ 和 $w_2 = aB$ 或 $w_2 = a$，其中 $A \in N, B \in N$ 和 $a \in T$，或 $S \to \lambda$

○ 诺姆·乔姆斯基（Avram Noam Chomsky，生于 1928 年） 乔姆斯基出生在费城，他的父亲是一位希伯来语的学者。乔姆斯基在宾夕法尼亚大学获得学士、硕士和博士学位。1950 年至 1951 年，他在宾夕法尼亚大学任教。1955 年受聘于麻省理工学院，开始他执教法语和德语的生涯。乔姆斯基现今领有麻省理工学院外国语和语言学的费拉雷·华德教授衔。他以其在语言学方面的基础贡献而著称，其中包括对文法的研究。乔姆斯基也以对政治的直言不讳而知名于世。

10.1.4 派生树

对上下文无关文法生成的语言，其派生可以用有序根树表示成图形，这样的树称为派生树或语法分析树，树根表示初始符；树的内部节点表示在派生过程中产生的非终结符，树的叶节点表示终结符。如果在派生过程中，用到了产生式 $A \rightarrow w$（其中 w 是一个词），则表示 A 的节点就有一些子节点，它们表示 w 中的每一个符号，并且从左到右排列。

例 11 对于本节引言所给的例子，构造派生 *the hungry rabbit eats quickly* 的派生树。

解 所求派生树如图 10-2 所示。

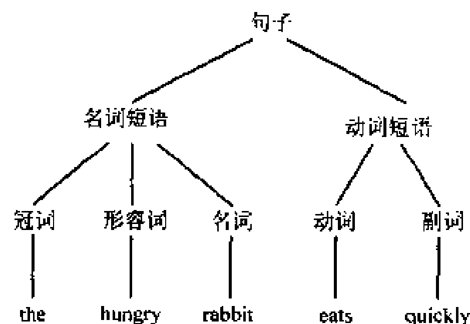


图 10-2 派生树

在许多应用中，都会遇到这样的问题：确定一个串是否在一个上下文无关文法生成的语言中，例如编译器的构造。下面例子指出了解决这样问题的两个方法。■

例 12 确定词 $cbab$ 是否在文法 $G = \{V, T, S, P\}$ 生成的语言中，其中，$V = \{a, b, c, A, B, C, S\}$，$T = \{a, b, c\}$，$S$ 为初始符，产生式为

$S \rightarrow AB$

$A \rightarrow Ca$

$B \rightarrow Ba$

$B \rightarrow Cb$

$B \rightarrow b$

$C \rightarrow cb$

$C \rightarrow b$

解 解决这个问题的一种办法是：从 S 出发，用一系列产生式试着派生出 $cbab$。因为只有一个产生式的左边是 S，故必须从 $S \Rightarrow AB$ 开始。下一步，用左边是 A 的唯一产生式 $A \rightarrow Ca$ 得到 $S \Rightarrow AB \Rightarrow CaB$。因为 $cbab$ 以符号 cb 开始，故我们使用产生式 $C \rightarrow cb$，这样就得到了 $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB$。最后，使用产生式 $B \rightarrow b$ 就可得到 $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB \Rightarrow cbab$。这种方法称为*自顶向下的语法分析*，因为它从第一个符号开始，一个接一个地用产生式来处理。

解决整个问题的另一个办法称为*自底向上的语法分析*，这种办法从后向前处理。因为 $cbab$ 是需要派生的串，故可以使用产生式 $C \rightarrow cb$，从而得到 $Cab \Rightarrow cbab$。再使用产生式 $A \rightarrow Ca$ 得到 $Ab \Rightarrow Cab \Rightarrow cbab$。由产生式 $B \rightarrow b$ 可得 $AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$。最后再用产生式 $S \rightarrow AB$，就可得到 $cbab$ 的一个完整的派生 $S \Rightarrow AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$。■

10.1.5 巴科斯-诺尔范式



有时候还用另一个方法来表示 2 型文法，这就是巴科斯-诺尔范式，这个方法是根据

约翰·巴科斯[⊖]和彼得·诺尔[⊖]命名的，约翰·巴科斯是它的发明人，彼得·诺尔则改进了它，并将之应用于程序设计语言 ALGOL 的规范说明。巴科斯－诺尔范式已被用来对许多程序设计语言（包括 Java）的语法规则进行归范说明。在 2 型文法中，产生式的左边都是单个非终结符。在巴科斯－诺尔范式中，将左边是同一个非终结符的所有产生式合并成一个式子，而不是将这些产生式都列出来。我们还用符号::＝代替→，将非终结符用< >括起来，并在一个式子里列出所有这些产生式的右边，用竖线将这些产生式分开。例如，产生式 $A\to Aa$，$A\to a$，$A\to AB$ 可以合并成⟨A⟩::＝⟨A⟩a | a | ⟨A⟩⟨B⟩。

例 13 本节引言描述了英语的一个子集，其对应文法的巴科斯－诺尔范式是什么？

解 这个文法的巴科斯－诺尔范式是：

⟨句子⟩::＝⟨名词短语⟩⟨动词短语⟩

⟨名词短语⟩::＝⟨冠词⟩⟨形容词⟩⟨名词⟩|⟨冠词⟩⟨名词⟩

⟨动词短语⟩::＝⟨动词⟩⟨副词⟩|⟨动词⟩

⟨冠词⟩::＝ *a* | *the*

⟨形容词⟩::＝ *large* | *hungry*

⟨名词⟩::＝ *rabbit* | *mathematician*

⟨动词⟩::＝ *eats* | *hops*

⟨副词⟩::＝ *quickly* | *wildly* ■

例 14 给出带符号十进制整数的产生式的巴科斯－诺尔范式（带符号整数是非负整数前面加上一个加号或减号）。

解 一个产生带符号整数的文法之巴科斯－诺尔范式为：

⟨带符号整数⟩::＝⟨符号⟩⟨整数⟩

⟨符号⟩::＝ + | −

⟨整数⟩::＝⟨数字⟩|⟨数字⟩⟨整数⟩

⟨数字⟩::＝0|1|2|3|4|5|6|7|8|9 ■

⊖ 约翰·巴科斯（John Backus，生于 1924 年） 巴科斯生于费城，在哥伦比亚大学获得理学学士和数学硕士学位。于 1950 年进入 IBM 当程序设计员。他参加了 IBM 的两种早期计算机的设计与开发。从 1954 年到 1958 年，他领导 IBM 的一个小组开发了 FORTRAN。巴科斯在 1958 年成为 IBM 沃森研究中心的职员。他是程序设计语言 ALGOL 设计委员会的一员，正是在此语言的设计过程中，他使用了现今称为巴科斯－诺尔范式的方法来描述此语言的语法。后来，巴科斯从事于集合簇的数学的研究和函数型程序设计的研究。巴科斯于 1963 年成为 IBM 的特别会员（fellow），于 1974 年获美国国家科学奖（National Medal of Science），于 1977 年获美国计算机协会（Association of Computing Machinery）颁发的具有崇高声誉的图灵奖。

⊖ 彼得·诺尔（Peter Naur，生于 1928 年） 彼得·诺尔生于哥本哈根附近的 Frederiksberg。孩提时代，诺尔就对天文学感兴趣。他不局限于观察天体，还计算彗星和小行星的轨道。诺尔毕业于哥本哈根大学，并于 1949 年获得学位。他在剑桥度过了 1950 和 1951 年，在此期间他用早期的计算器来计算彗星和行星的运动。回到丹麦后，他虽然继续从事于天文学的研究，但已经与计算紧密联系起来了。在 1955 年，他作为顾问参与丹麦的第一个计算机的建造。在 1959 年，诺尔从天文学转入计算的研究，并将计算的研究作为专职工作。作为一个专职计算机科学家，他的第一个工作是参加程序设计语言 ALGOL 的开发。从 1960 至 1967 年，他继续从事于 ALGOL 和 COBOL 编译器的研究。在 1969 年，他成为哥本哈根大学的计算机科学教授，在那里，他在程序设计方法学领域辛勤耕耘着。

练习

练习 1～3 中的文法是：初始符为句子，终结符集 $T=\{the, sleepy, happy, tortoise, hare, passes, runs, quickly, slowly\}$，非终结符集 $N=\{$名词短语，及物动词短语，不及物动词短语，冠词，形容词，名词，动词，副词$\}$，产生式为

句子→名词短语　及物动词短语　名词短语

句子→名词短语　不及物动词短语

名词短语→冠词　形容词　名词

名词短语→冠词　名词

及物动词短语→及物动词

不及物动词短语→不及物动词　副词

不及物动词短语→不及物动词

冠词→*the*

形容词→*sleepy*

形容词→*happy*

名词→*tortoise*

名词→*hare*

及物动词→*passes*

不及物动词→*runs*

副词→*quickly*

副词→*slowly*

1. 用产生式集证明下列每个句子都是有效句子：
   a) *the happy hare runs*
   b) *the sleepy tortoise runs quickly*
   c) *the tortoise passes the hare*
   d) *the sleepy hare passes the happy tortoise*
2. 除了练习 1 中的有效句子外，再给出五个有效句子。
3. 证明：*the hare runs the sleepy tortoise* 不是有效句子。

*4. 设 $V=\{S,A,B,a,b\}$，$T=\{a,b\}$。当产生式集为下列情形之一时，求文法 $\{V,T,S,P\}$ 生成的语言。
   a) $S\to AB$，$A\to ab$，$B\to bb$
   b) $S\to AB$，$S\to aA$，$A\to a$，$B\to ba$
   c) $S\to AB$，$S\to AA$，$A\to aB$，$A\to ab$，$B\to b$
   d) $S\to AA$，$S\to B$，$A\to aaA$，$A\to aa$，$B\to bB$，$B\to b$
   e) $S\to AB$，$A\to aAb$，$B\to bBa$，$A\to\lambda$，$B\to\lambda$

5. 用例 5 所给的文法构造 $0^3 1^3$ 的派生。
6. 证明：例 5 所给的文法生成集合 $\{0^n 1^n \mid n=0,1,2,\cdots\}$。

7. a) 用例 6 中的文法 G_1 构造 $0^2 1^4$ 的派生。

   b) 用例 6 中的文法 G_2 构造 $0^2 1^4$ 的派生。

8. a) 证明：例 6 中的文法 G_1 生成集合$\{0^m 1^n \mid m, n = 0, 1, 2, \cdots\}$。

   b) 证明：例 6 中的文法 G_2 生成同一个集合。

9. 用例 7 所给的文法构造 $0^2 1^2 2^2$ 的派生。

*10. 证明：例 7 所给的文法生成集合$\{0^n 1^n 2^n \mid n = 0, 1, 2, \cdots\}$。

*11. 求下列语言的短语结构文法

   a) 包含偶数个 0 但没有 1 的所有二进制串构成的集合。

   b) 由 1 后面跟奇数个 0 的所有二进制串构成的集合。

   c) 包含偶数个 0 和偶数个 1 的所有二进制串构成的集合。

   d) 包含 10 个以上 0 但没有 1 的所有二进制串构成的集合。

   e) 所包含 0 的个数多于 1 的个数的所有二进制串构成的集合。

   f) 包含相同个数的 0 和 1 的所有二进制串构成的集合。

   g) 包含不同个数的 0 和 1 的所有二进制串构成的集合。

12. 构造生成下列集合的短语结构文法：

   a) $\{01^{2n} \mid n \geq 0\}$

   b) $\{0^n 1^{2n} \mid n \geq 0\}$

   c) $\{0^n 1^m 0^n \mid m \geq 0, n \geq 0\}$

13. 设 $V = \{S, A, B, a, b\}$，$T = \{a, b\}$。若产生式集 P 为下列集合时，问文法$G = \{V, T, S, P\}$是否为 0 型但不是 1 型文法？是否为 1 型但不是 2 型文法？或是否为 2 型但不是 3 型文法？

   a) $S \to aAB$，$A \to Bb$，$B \to \lambda$

   b) $S \to aA$，$A \to a$，$A \to b$

   c) $S \to ABa$，$AB \to a$

   d) $S \to ABA$，$A \to aB$，$B \to ab$

   e) $S \to bA$，$A \to B$，$B \to a$

   f) $S \to aA$，$aA \to B$，$B \to aA$，$A \to b$

   g) $S \to bA$，$A \to b$，$S \to \lambda$

   h) $S \to AB$，$B \to aAb$，$aAb \to b$

   i) $S \to aA$，$A \to bB$，$B \to b$，$B \to \lambda$

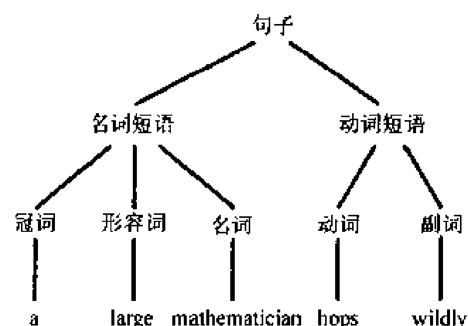
   j) $S \to A$，$A \to B$，$B \to \lambda$

14. 回文是从前向后读和从后向前读都一样的串，也就是 $w = w^R$ 这样的串 w，其中 w^R 是串 w 的倒转。试求一个上下文无关的文法，使得其生成的集合是字母表$\{0, 1\}$上的所有回文。

*15. 设 G_1 和 G_2 是两个上下文无关的文法，它们生成的语言分别为$L(G_1)$和 $L(G_2)$。试证：对于下列每个集合，都有一个上下文无关文法来生成下列集合。

   a) $L(G_1) \cup L(G_2)$　　b) $L(G_1)L(G_2)$　　c) $L(G_1)^*$

16. 求用下图派生树构造的串。



17. 构造练习 1 中的句子的派生树。
18. 设 G 是一个文法，其中 $V=\{a,b,c,S\}$，$T=\{a,b,c\}$，初始符号为 S，产生式为 $S\to abS$，$S\to bcS$，$S\to bbS$，$S\to a$，$S\to cb$。构造下列串的派生树。

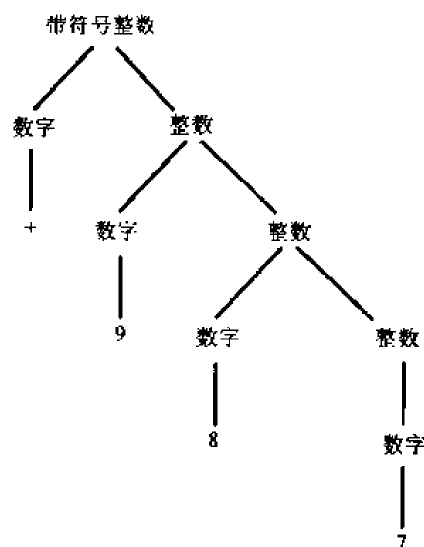
    a) *bcbba*　　b) *bbbcbba*　　c) *bcabbbbbcb*

*19. 对于下列每个串，用自顶向下的语法分析方法，确定其是否属于练习 12 中的文法生成的语言。

    a) *baba*　　b) *abab*

    c) *cbaba*　　d) *bbbcba*

*20. 对于练习 19 中的串，用自底向上的语法分析方法，确定其是否属于练习 12 中的文法生成的语言。



21. 用例 14 所给的文法构造 −109 的派生树。
22. a) 如果一个文法的产生式由下列巴科斯－诺尔范式给出，这些产生式是什么？

    〈表达式〉::=(〈表达式〉)|〈表达式〉+〈表达式〉|〈表达式〉*〈表达式〉|〈变元〉

    〈变元〉::=x|y

    b) 求此文法中$(x*y)+x$ 的派生树。
23. a) 构造一个短语结构文法，使其生成如下所有带符号十进制数：这些数由符号（+或－）、非负整数和十进制小数三部分构成，且十进制小数部分或者是空串，或者是小数点后面跟一个正整数，其中，整数的开始部分允许有一些 0。

    b) 给出这个文法的的巴科斯－诺尔范式。

    c) 构造此文法中 −31.4 的派生树。
24. a) 构造一个短语结构文法，使其生成所有形如 a/b 的分数构成的集合，其中 a 为带符号十进制数，b 是正整数。

    b) 给出这个文法的的巴科斯－诺尔范式。

    c) 构造此文法中 +311/17 的派生树。
25. 设 G 是一个文法，R 是一个关系，有序对$(w_0,w_1)\in R$ 当且仅当 w_1 可以从 w_0 在 G 中直接派生出来。求 R 的自反传递闭包。

10.2 带输出的有限状态机

10.2.1 引言

许多种类的机器，包括计算机的某些部件，都可以用有限状态机作为模型。经常用来作为模型的有限状态机也有多种形式，但所有这些形式都包括一个有限的状态集合

(其中有一个指定的初始状态)、一个输入字母表和一个转移函数（对每个由状态和输入构成的对指定下一个状态)。本节将讨论带输出的有限状态机，说明怎样用有限状态机来构造下列机器的模型：自动售货机、输入延迟机、整数加法器和用来确定一个二进制串是否包含一个特定模式的机器。

在给出形式定义之前，先来说明怎么建立自动售货机的模型。自动售货机可以接受 5 分、1 角和 25 分硬币。如果 30 分或更多硬币被投到机器里，则机器立刻退出超过 30 分的部分。如果顾客投放了 30 分且超出部分已被退还，则顾客可以按橙色按钮得到一筒橘子汁，或者按红色按钮得到一筒苹果汁。可以如下描述这个机器是怎么工作的：详细描述它的状态，且说明它在接受输入后怎么改变状态，还要说明对输入和当前状态的各种组合，它所产生的输出。

这个机器可能处于 7 种状态 $s_i(i=0,1,2,\cdots,6)$，其中状态 s_i 指机器已经收集了 $5i$ 分。机器以表示收集了 0 分的状态 s_0 开始。输入可能是：5 分、1 角、25 分、橙色钮(O)或红色钮(R)。输出可能是：空(n)、5 分、1 角、15 分、20 分、25 分、一筒橘子汁或一筒苹果汁。

下面的例子说明了此机器的模型是怎么工作的。假设一个学生先投入了 1 角，又投入了 25 分，得到了 5 分的找赎，然后按橙色按钮就得到一筒橘子汁。机器从状态 s_0 开始。它的第一个输入是 10 分，这就将机器的状态改变为 s_2，但没有输出。第二个输入是 25 分，这将状态从 s_2 改变为 s_6，并返回 5 分作为输出。下一个输入是橙色按钮，它将状态从 s_6 改回到 s_0（因为机器返回到初始状态)，并送出一筒橘子汁作为输出。

可以将机器的所有这些状态变化和输出用一个表来表示。为此，对状态和输入的每个组合，我们都需要指明下一个状态和产生的输出。表 10-2 对每对状态和输入都指明了转移和输出。

表 10-2　自动售货机的状态表

状　态	下一个状态					输　出				
	输　入					输　入				
	5	10	25	O	R	5	10	25	O	R
s_0	s_1	s_2	s_5	s_0	s_0	n	n	n	n	n
s_1	s_2	s_3	s_6	s_1	s_1	n	n	n	n	n
s_2	s_3	s_4	s_6	s_2	s_2	n	n	5	n	n
s_3	s_4	s_5	s_6	s_3	s_3	n	n	10	n	n
s_4	s_5	s_6	s_6	s_4	s_4	n	n	15	n	n
s_5	s_6	s_6	s_6	s_5	s_5	n	5	20	n	n
s_6	s_6	s_6	s_6	s_0	s_0	5	10	25	OJ	AJ

说明机器动作的另一个方法是使用带标号边的有向图，其中：状态表示为小圈，边表示转移，并用输入和这个转移产生的输出对边进行标号。自动售货机的有向图如图 10-3 所示。

10.2.2　带输出的有限状态机

现在给出带输出的有限状态机的形式定义。

定义 1　有限状态机 $M=(S,I,O,f,g,s_0)$ 由如下部分组成：一个有限的状态集合 S，一个有限的输入字母表 I，一个有限的输出字母表 O，一个转移函数 f，f 为每个状态和输入对指派一个新状态，一个输出函数 g，g 为每个状态和输入对指派一个输出，还有一个初始状态 s_0。

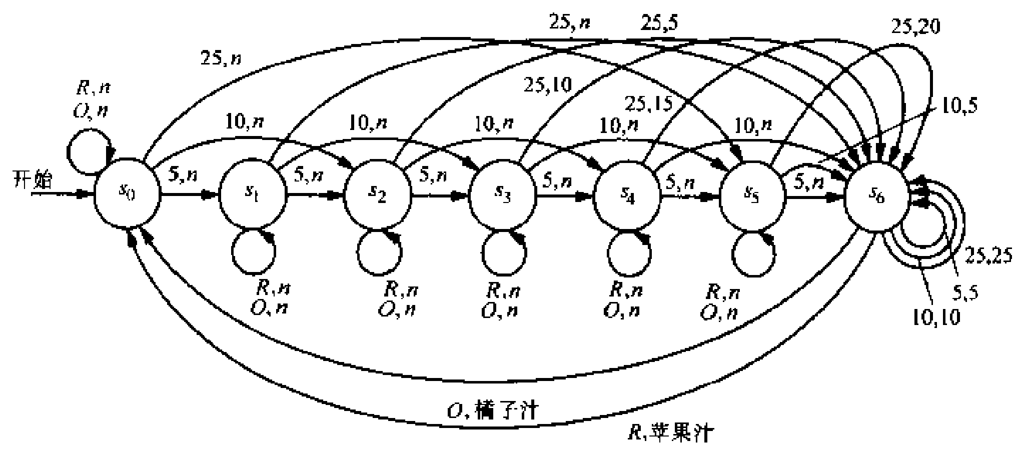


图 10-3 一个自动售货机

设$M=(S,I,O,f,g,s_0)$是一个有限状态机，可以用状态表来表示状态函数和输出函数的值。在本节引言中，我们已经构造了自动售货机的状态表。

例 1 表 10-3 的状态表描述了一个有限状态机，其中$S=\{s_0,s_1,s_2,s_3\}$，$I=\{0,1\}$，且$O=\{0,1\}$。转移函数 f 的值在前两列给出，输出函数 g 的值在后两列给出。 ■

表示有限状态机的另一种方法是用**状态图**，这是一个带标号边的有向图。在这个图中，状态由圈表示，转移由带标号的输入和输出对箭头表示。

例 2 构造状态表如表 10-3 所示的有限状态机的状态图。

解 这个机器的状态图如图 10-4 所示。 ■

表 10-3

状态	f		g	
	输入		输入	
	0	1	0	1
s_0	s_1	s_0	1	0
s_1	s_3	s_0	1	1
s_2	s_1	s_2	0	1
s_3	s_2	s_1	0	0

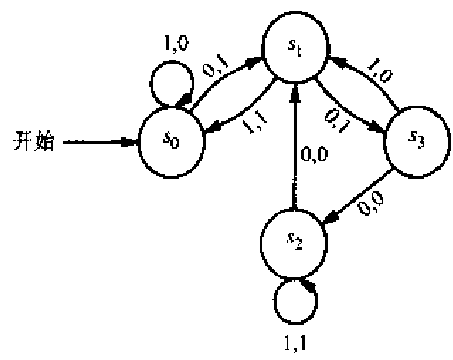


图 10-4 表 10-3 所示有限状态机的状态图

例 3 构造状态图如图 10-5 所示的有限状态机的状态表。

解 这个机器的状态表如表 10-4 所示。 ■

一个输入串使初始状态经历一系列状态，这些状态都是由转移函数确定的。当我们（从左向右）一个符号一个符号地读输入串的时候，每个输入符号都使机器从一个状态变为另一个状态。因为每个转移产生一个输出，故一个输入串产生一个输出串。

设输入串为 $x=x_1x_2\cdots x_k$。则读这个输入使得机器从状态 s_0 变为状态 s_1，其中 $s_1=f(s_0,x_1)$，然后变为状态 s_2，其中 $s_2=f(s_1,x_2)$，等等，以状态$s_k=f(s_{k-1},x_k)$结束。这个转移序列就产生了输出串 $y=y_1y_2\cdots y_k$，其中 $y_1=g(s_0,x_1)$是对应于从 s_0 到 s_1 的转移的输

出，$y_2=g(s_1,x_2)$是对应于从 s_1 到 s_2 的转移的输出，等等。一般地，$y_j=g(s_{j-1},x_j)$，$j=1$，2，…，k。这样我们可以将输出函数 g 的定义扩展到输入串，即定义$g(x)=y$，其中 y 是对应输入串 x 的输出。在许多应用中，这个记号都很有用。

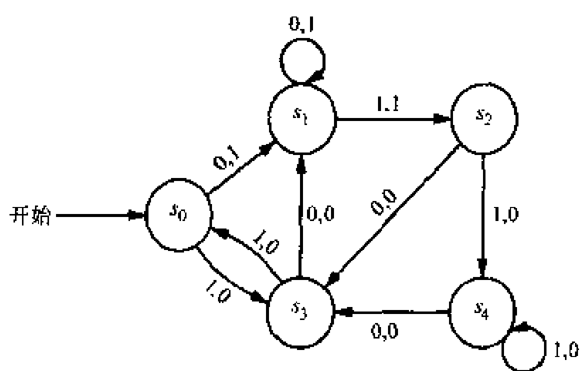


图 10-5　一个有限状态机

表　10-4

状态	f 输入 0	f 输入 1	g 输入 0	g 输入 1
s_0	s_1	s_3	1	0
s_1	s_1	s_2	1	1
s_2	s_3	s_4	0	0
s_3	s_1	s_0	0	0
s_4	s_3	s_4	0	0

例 4　对于图 10-5 表示的有限状态机，求其对输入串 101011 生成的输出串。

解　输出是 001000。状态和输出的逐次变化如表 10-5 所示。■

表　10-5

输入	1	0	1	0	1	1	—
状态	s_0	s_3	s_1	s_2	s_3	s_0	s_3
输出	0	0	1	0	0	0	—

例 5　单位延迟机是许多电子装置中的一个重要部件，它将输入串延迟一定时间量后输出。怎么构造一个有限状态机使其将输入串延迟一个单位时间呢？即：对于输入的二进制串 $x_1x_2\cdots x_k$，怎么才能输出二进制串 $0x_1x_2\cdots x_{k-1}$?

解　可以如下构造一个延迟机：它有两种可能的输入，即 0 和 1，它还必须有一个初始状态 s_0。因为它还要记住前一个输入是 0 还是 1，所以还需要另外两个状态 s_1 和 s_2，使得如果前一个输入是 1，则机器处于状态 s_1，如果前一个输入是 0，则机器处于状态 s_2。从 s_0 出发的第一个转移产生输出 0，从 s_1 出发的每个转移都产生输出 1，从 s_2 出发的每个转移都产生输出 0。则对应于输入串 $x_1x_2\cdots x_k$ 的输出是这样一个串：以 0 开始，后面跟 x_1，再跟 x_2，…，最后以 x_{k-1}结束。这个机器的状态图如 10－6 所示。■

例 6　试构造一个有限状态机，使其利用整数的二进制展开式将两个整数相加。

解　一般按如下过程将$(x_n\cdots x_1x_0)_2$ 和$(y_n\cdots y_1y_0)_2$ 相加（如 2.4 节所描述）：首先，将数位 x_0 和 y_0 相加，产生和位 z_0 与进位 c_0，且此进位要么是 0，要么是 1；然后将数位 x_1，y_1 连同进位 c_0 一起相加，产生和位 z_1 和进位 c_1；将这个过程一直进行下去；第 n 步是将 x_n，y_n 连同前一个进位 c_{n-1}一起相加，产生和位 z_n 和进位 c_n，c_n 也就是和位 z_{n+1}。

只用两个状态就能构造执行这个加法的有限状态机。为简单起见，假设两个初始位 x_n 和 y_n 都是 0（否则，必须对和位 z_{n+1}作特殊安排）。我们用初始状态 s_0 表示前一个进位是 0（或者是最右边数位的加法），用另一个状态 s_1 表示前一个进位是 1。因为这个机器的输入是一对二进制数，所以只有四种可能的输入。这四种可能的输入为：00（两个二进制数都是 0）、01（第一个二进制数为 0，第二个为 1）、10（第一个二进制数为 1，第二个为 0）和 11（两个二进制数都是 1）。转移和输出是根据下面两个因素来构造的：一个是输入所表示的两个二进制数的和，另一个是状态所表示的进位。例如：当机器处于状态 s_1 且所接受的输入

是 01 时，则下一个状态是 s_1 且输出是 0，因为所产生的和是 $0+1+1=(10)_2$。此机器的状态图如图 10-7 所示。■

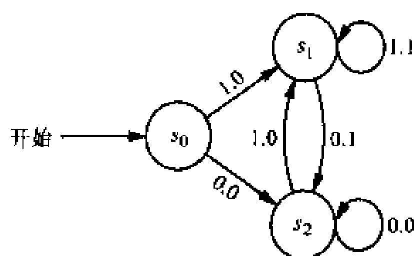


图 10-6 一个单位延迟机

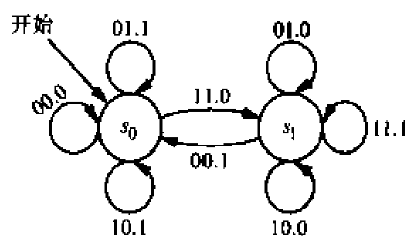


图 10-7 一个做加法的有限状态机器

例 7 在某种编码方法中，当一个信息中出现了 3 个连续的 1，则信息接收器就知道已经发生了一个传送错误。试构造一个有限状态机，使得它输出 1，当且仅当它所接收的最后 3 位都是 1。

解 这个机器需要三个状态。初始状态 s_0 记住前一个输入值（如果存在话）不是 1；状态 s_1 记住前一个输入值是 1，但再前一个输入（如果存在的话）不是 1；状态 s_2 记住前两个输入值都是 1。输入一个 1 将状态 s_0 变为 s_1，因为机器现在读到的是单个的 1，而不是两个连续的 1；它将 s_1 变为 s_2，因为它现在读到了两个连续的 1；它还将 s_2 变为 s_2 本身，因为它已经至少读到了两个连续的 1。输入一个 0 将每个状态都变为 s_0，因为这打断了任何由连续 1 构成的串。如果现在机器所读的是 1，则由 s_2 到 s_2 自身的转移所产生的输出为 1，因为此状态与输入的组合表明机器已经读到了 3 个连续的 1。其他情形的输人都是 0。此机器的状态图如图 10-8 所示。■

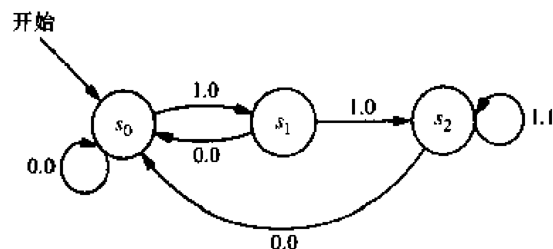


图 10-8 一个有限状态机，它输出 1 当且仅当所读的输入串以 111 结尾

图 10-8 所示的机器是语言识别器的一个例子，因为它输出 1 当且仅当它所读到的输入串具有某种特定的性质。语言识别是有限状态机的一个重要应用。

有限状态机的类型： 为建立计算机的模型，人们开发了许多种不同的有限状态机。本节给出了一类有限状态机的定义，在这种类型的机器中，输出与状态间的转移相对应，这种类型的机器称作*米利机*，因为它是由米利（G. H. Mealy）在 1955 年首先研究的。还有另外一类重要的带输出的有限状态机，其输出仅仅由状态确定，这种类型的有限状态机称为*摩尔机*，因为它是摩尔（E. F. Moore）在 1956 年提出的。本节结尾有一系列练习讨论摩尔机。

例 7 说明了怎么用米利机来识别语言，但我们通常用另一种不带输出的有限状态机来识别语言。不带输出的有限状态机也称为有限状态自动机，它有一个由终结状态组成的集合，它识别一个串当且仅当能够将初始状态变为一个终结状态。10.3 节将讨论这种类型的有限状态机。

练习

1. 画出具有下列状态表的有限状态机的状态图。

a)

状态	f		g	
	输入		输入	
	0	1	0	1
s_0	s_1	s_0	0	1
s_1	s_0	s_2	0	1
s_2	s_1	s_1	0	0

b)

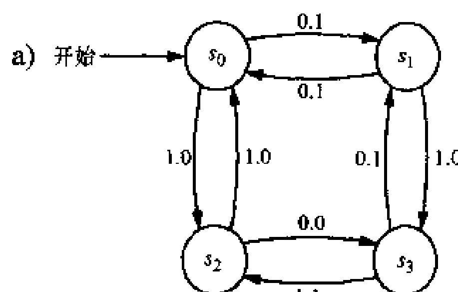
状态	f		g	
	输入		输入	
	0	1	0	1
s_0	s_1	s_0	0	0
s_1	s_2	s_0	1	1
s_2	s_0	s_3	0	1
s_3	s_1	s_2	1	0

c)

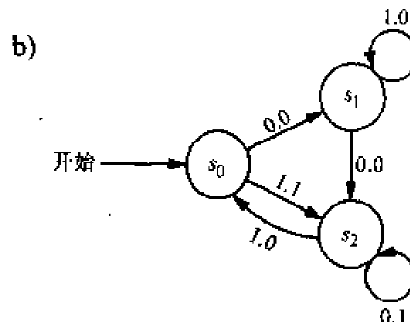
状态	f		g	
	输入		输入	
	0	1	0	1
s_0	s_0	s_4	1	1
s_1	s_0	s_3	0	1
s_2	s_0	s_2	0	0
s_3	s_1	s_1	1	1
s_4	s_1	s_0	1	0

2. 给出具有下列状态图的有限状态机的状态表。

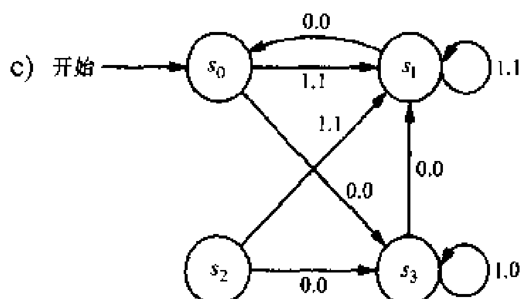
a)



b)



c)



3. 在例 2 所给的有限状态机中，对于下列每个输入串，试确定其输出。

a) 0111　　b) 11011011　　c) 01010101010

4. 在例 3 所给的有限状态机中，对于下列每个输入串，试确定其输出。

a) 0000　　b) 101010　　c) 11011100010

5. 试构造一个有限状态机作为下列饮料机的模型：饮料机接受五分、一角和 25 分的硬币，它一直到接受了 35 分钱币时才开始找回零钱，退出超过 35 分的所有钱币。然后顾客就可以按一些按钮，得到一筒可乐，或一瓶软饮料，或一瓶姜汁啤酒。

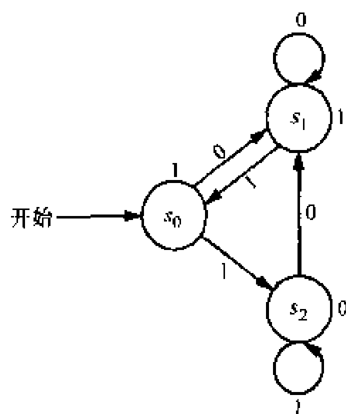
6. 试构造一个有限状态机作为下列售报机的模型：它有一个门，此门只在下列两种情形下才可打开：一是放入了 3 个一角硬币（和任意数量的其他硬币），二是放入了一个 25 分的硬币和一个 5 分的硬币（和任意数量的其他硬币）。一旦门能够被打开，顾客就打开门，取出一份报纸，再关上门。不管塞进去多少额外的钱币，机器都不找回零钱。下一个顾客重新开始时也不能使用上一位多余的钱。

7. 构造一个有限状态机，将输入延迟两位，且以00作为输出的头两位。
8. 构造一个有限状态机，对输入串每隔一位改变一次值，且从第二位开始。但保持其他位不变
9. 构造一个有限状态机来模拟计算机的登录过程：用户首先输入用户标识码，然后输入口令；用户标识码和口令分别被看作是一个输入；如果输入的口令不对，则要求用户重新输入用户标识码。
10. 构造一个有限状态机来模拟密码锁，此锁包含数1到40，它只有在输入正确的组合时才能被打开，正确组合是：10次右，8次左，37次右。每个输入都是"一个数、旋转方向、在此方向旋转锁的次数"构成的三元组。
11. 构造一个有限状态机来模拟下列道路收费机：放入25分钱币之后此机器将打开一个门。可以使用面额为5分、一角和25分的硬币。不找零钱，多于25分的超额部分也不提供下一位驾驶者使用。
12. 构造一个有限状态机：当读取的输入符号所代表的数能够被3整除时，输出1；否则输出0。
13. 构造一个有限状态机：确定在输入串中当前所读取的最后一个符号是否为1，且倒数第三个符号是否为0。
14. 构造一个有限状态机：确定到目前为止所读取的输入串中，其结尾是否有至少5个连续的1。
15. 构造一个有限状态机：确定到目前为止所读取的输入中，其最后的八个字符是否为computer。输入可能是任意的英文字母。

摩尔机 $M=(S,I,O,f,g,s_0)$由下列六部分构成：有限状态集S；输入字母表I；输出字母表O；转移函数f，它将每个由状态和输入组成的对映射为下一个状态；输出函数g，它对每个状态指定一个输出；初始符号s_0。摩尔机可以用状态表来表示，也可以用状态图来表示。状态表列出对应于每个状态和输入对的转移，以及对每个状态的输出。状态图画出状态、状态间的转移以及状态的输出。在状态图中，转移用带标号的箭头和输入表示，输出写在状态的旁边。

16. 构造具有下列状态表的摩尔机的状态图。
17. 构造具有下列状态图的摩尔机的状态表。对每个输入串，摩尔机都产生一个输出串。特

状态	f 输入 0	f 输入 1	g
s_0	s_0	s_2	0
s_1	s_3	s_0	1
s_2	s_2	s_1	1
s_3	s_2	s_0	1



别对应于输入串 $a_1a_2\cdots a_k$ 的输出是 $g(s_0)g(s_1)\cdots g(s_k)$，其中 $s_i=f(s_{i-1},a_i)$，$i=1,2,\cdots,k$。

18. 对于下列每个输入串，求练习 16 中的摩尔机所生成的输出串。
    a) 0101　　b) 11111　　c) 11101110111
19. 对于练习 18 中的每个输入串，求练习 17 中的摩尔机所生成的输出串。
20. 构造一个摩尔机：只要读取的输入符号的个数能够被 4 整除时，就输出 1。
21. 构造一个摩尔机，使其能够判断输入串是包含偶数个 1 还是奇数个 1。如果输入串中有偶数个 1，则输出 1；如果输入串中有奇数个 1，则输出 0。

10.3 不带输出的有限状态机

10.3.1 引言

有限状态机的最重要应用之一是语言识别。在设计和构造程序设计语言的编译器时，这个应用起着根本性的作用。在 10.2 节中，我们说明了可以用带输出的有限状态机来识别语言，方法是当读取的输入串在语言中时输出 1，否则输出 0。但是，还有一些其他类型的有限状态机，它们是为识别语言而专门设计的。这些机器不产生输出，但有终结状态。一个串能被它识别，当且仅当它将初始状态带到终结状态。

10.3.2 串的集合

在讨论不带输出的有限状态机之前，先介绍一些关于串的集合的重要背景材料。这里定义的运算将广泛用于有限状态机识别的语言的讨论。

定义 1　设 V 是一个词汇表，A、B 是 V^* 的子集。A 和 B 的连接是所有形如 xy 的串构成的集合，记为 AB，其中 x 是 A 中的串，y 是 B 中的串。

例 1　设 $A=\{0,11\}$，$B=\{1,10,110\}$。求 AB 和 BA。

解　集合 AB 包括所有 A 中串和 B 中串的连接，故 $AB=\{01,010,0110,111,1110,11110\}$。集合 BA 包括所有 B 中串和 A 中串的连接，故 $BA=\{10,111,100,1011,1100,11011\}$。 ■

注意，对于字母表 V 和 V^* 的子集 A 与 B，$AB=BA$ 不一定成立，如例 1 所示。

由两个串集合连接的定义还可以定义 $A^n(n=0,1,2,\cdots)$。其递归定义如下：

$$A^0=\{\lambda\}$$

$$A^{n+1}=A^nA,\ n=0,1,2,\cdots$$

例 2　设 $A=\{1,00\}$。当 $n=1,2,3$ 时求 A^n。

解　易知：$A^0=\{\lambda\}$，$A^1=A^0A=\{\lambda\}A=\{1,00\}$。为求 A^2，取 A 中元素对的连接。从而 $A^2=\{11,100,001,0000\}$。为求 A^3，取 A^2 和 A 中的元素进行连接，由此得到 $A^3=\{111,1100,1001,10000,0011,00100,00001,000000\}$。 ■

定义 2　设 A 是 V^* 的一个子集。A 的克莱因[○]闭包是 A 中任意多个串的连接组成的集合，记为 A^*，即 $A^*=\bigcup_{k=0}^{\infty}A^k$。

例 3 求集合$A=\{0\}, B=\{0,1\}, C=\{11\}$的克莱因闭包。

解 A 的克莱因闭包是 0 与自己的任意多次连接，故 $A^*=\{0^n \mid n=0,1,2,\cdots\}$。$B$ 的克莱因闭包是任意多个串的连接，但这些串只能是 0 或 1，因而这个闭包是字母表 $V=\{0,1\}$上的所有串，即 $B^*=V^*$。最后，C 的克莱因闭包是 11 与自己的任意多次连接，所以，C^*是由偶数个 1 组成的串的集合，即 $C^*=\{1^{2n} \mid n=0,1,2,\cdots\}$。 ■

10.3.3 有限状态自动机

现在给出不带输出的有限状态机的定义，这样的机器也叫*有限状态自动机*（finite-state automata），这也是在本节中将使用的术语。（注：automata 的单数形式是 automaton。）这些机器与 10.2 节中研究的有限状态机不同，它们不产生输出，但它们有一个终结状态集合。我们将看到，它们识别的字符串是将初始状态变为终结状态的输入串。

定义 3 *有限状态自动机* $M=(S,I,f,s_0,F)$由下列五部分组成：一个有限的*状态*集合 S，一个有限的*输入字母表* I，一个*转移函数* f（f 为每个状态和输入对指派下一个状态），一个*初始状态* s_0，和一个由终结状态构成的 S 的子集 F。

有限状态自动机也可用状态表或状态图来表示。在状态图中，终结状态用双圈表示。

例 4 构造有限状态自动机$M=(S,I,f,s_0,F)$的状态图，其中 $S=\{s_0,s_1,s_2,s_3\}, I=\{0,1\}, F=(s_0,s_1)$，转移函数 f 如表 10-6 所示。

解 所求的状态图如图 10-9 所示。注意：输入 0 和 1 都将 s_2 变为 s_0，所以从 s_2 到 s_0 的边上有 0 和 1。 ■

表 10-6

状态	f 输入 0	f 输入 1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

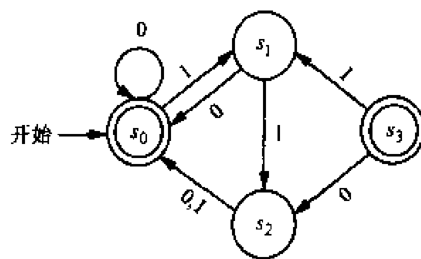


图 10-9 一个有限状态自动机的状态图

可以扩展转移函数，使其对所有状态与串的对都有值。设 $x=x_1x_2\cdots x_k$ 是 I^* 中的一个串，则$f(s_1,x)$是这样得到的状态：以状态 s_1 开始，对 x 从左到右连续地使用其每个符号。从 s_1 我们用$f(s_1,x)=f(s_k,x_k)$进入状态 $s_2=f(s_1,x_1)$，然后进入状态 $s_3=f(s_2,x_2)$，等等。

⊖ 克莱因（Stephen Cole Kleene，1909—1994） 克莱因生于美国康涅狄克州的哈特福德。他的母亲艾丽丝·科尔（Alice Lena Cole)是一位诗人，他的父亲古斯塔夫·克莱因(Gustav Adolph Kleene)是一位经济学教授。克莱因曾就读于 Amherst 学院，1934 年在普林斯顿大学获博士学位，导师是著名逻辑学家丘奇(Alonzo Church)。1935 年，克莱因成为威斯康星大学的教员，除了有几次离开（包括去普林斯顿高等研究所）外，他一直待在那里。二战期间，他成为美国海军预备役军官学校的航海教师，后来成为海军研究实验室的主任。克莱因研究可计算性和可判定性问题，对递归函数论作出了重要贡献，并且证明了自动机理论中的一个中心结果。他曾是威斯康星大学的数学研究中心的代理主任和文理学院的院长。他还曾经学习过博物学，发现了一族以前没有描述过的蝴蝶，这族蝴蝶就他的名字命名。他还是一个热心的徒步旅行者和登山者。克莱因还因能机敏地讲奇闻轶事而闻名，他的大嗓门在几间办公室之外都能听到。

称串 x 可以被机器$M=(S,I,f,s_0,F)$识别或接受，如果 x 将初始状态变为一个终结状态，即$f(s_0,x)$是 F 中的一个状态。机器 M 识别（或接受）的语言是 M 识别的所有串的集合，记为$L(M)$。如果两个有限状态自动机识别相同的语言，则称它们是等价的。

例 5　求图 10-10 表示的有限状态自动机 M_1，M_2 和 M_3 所识别的语言。

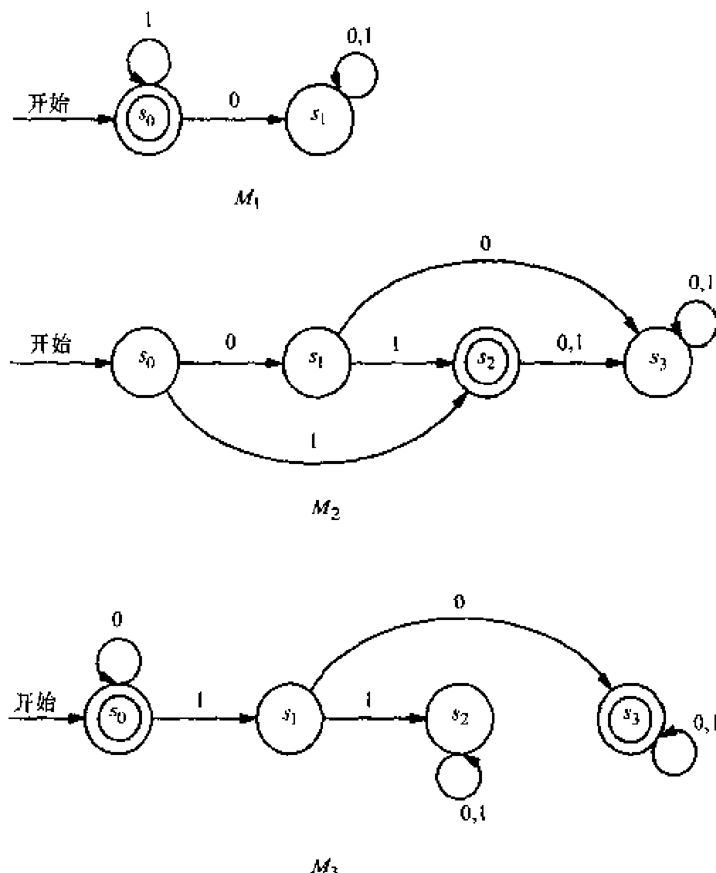


图 10-10　一些有限状态自动机

解　M_1 只有一个终结状态 s_0，而将 s_0 变为自身的串是由 0 个、1 个或更多连续 1 组成的串。所以$L(M_1)=\{1^n \mid n=0,1,2,\cdots\}$。

M_2 只有一个终结状态 s_2，而将 s_0 变为 s_2 的串只有 1 和 01，所以$L(M_1)=\{1,01\}$。

M_3 的终结状态有 s_0 和 s_3，将 s_0 变为自身的串有 λ，0，00，000，…，即由零个以上（包括零个）连续的 0 构成的串。将 s_0 变为 s_3 的串只有这样的串：开头是零个以上（包括零个）连续的 0 构成的串，接着是 10，然后是任意的串。故$L(M_3)=\{0^n,0^n10x \mid n=0,1,2,\cdots,x$ 是任意的串$\}$。 ■

到目前为止所讨论的有限状态自动机都是**确定型的**，因为对每对状态和输入值，转移函数只给出唯一的下个一状态。还有一种重要的有限状态自动机，它对每对输入值和状态，有多个可能的下一个状态，这样的机器称为**非确定型的**。非确定型有限状态自动机在判断哪些语言可以由有限状态自动机识别中非常重要。

定义 4　**非确定型有限状态自动机**$M=(S,I,f,s_0,F)$由下列五部分组成：一个状态的集合 S，一个输入字母表 I，一个转移函数 f（f 为每个状态和输入对指派一个状态集合），

一个初始状态 s_0，和一个由终结状态构成的 S 的子集 F。

非确定型有限状态自动机也可用状态表和状态图来表示。在状态表中，对每对状态和输入值，列出所有可能的下一个状态。在状态图中，从一个状态到每个可能的下一个状态，都画一个边，这个边的标号是导致这个转移的一个或多个输入。

例 6 求状态表如表 10-7 所示的非确定型有限状态自动机的状态图。终结状态为 s_2 和 s_3。

解 这个自动机的状态图如图 10-11 所示。 ■

表 10-7

状态	f 输入 0	f 输入 1
s_0	s_0，s_1	s_3
s_1	s_0	s_1，s_3
s_2		s_0，s_2
s_3	s_0，s_1，s_2	s_1

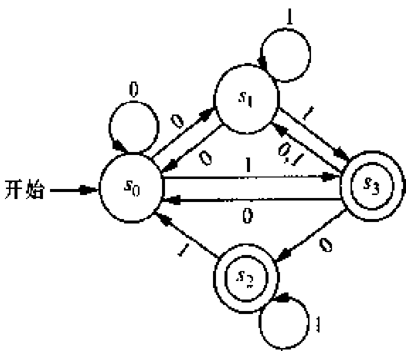


图 10-11 其状态表为表 10-7 的非确定型有限状态自动机

例 7 求状态图如图 10-12 所示的非确定型有限状态自动机的状态表。

解 这个自动机的状态表如表 10-8 所示。 ■

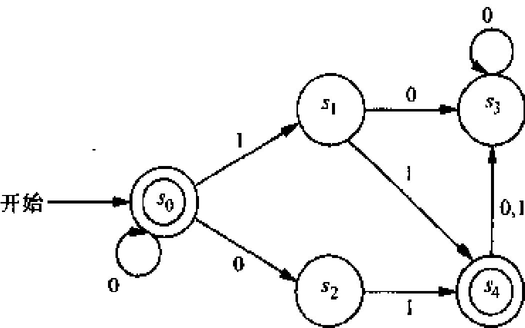


图 10-12 一个非确定型有限状态自动机

表 10-8

状态	f 输入 0	f 输入 1
s_0	s_0，s_2	s_1
s_1	s_3	s_4
s_2		s_4
s_3	s_3	
s_4	s_3	s_3

非确定型有限状态自动机怎么识别串 $x=x_1x_2\cdots x_k$ 呢？第一个输入符号 x_1 将初始状态 s_0 变为状态集合 s_1。下一个输入符号 x_2 将 s_1 中的每个状态都变为一个状态集合。设 s_2 是这些集合的并。将这个过程继续下去，在一个步骤中，对上一个步骤产生的每个状态和当前的输入符号，都要求其产生的状态。使用 x 从 s_0 所能得到的状态中，如果有一个终结状态，我们就识别（或接受）串 x。非确定型有限状态自动机所识别的语言是这个自动机所识别的所有串的集合。

例 8 求图 10-12 所示的非确定型有限状态自动机所识别的语言。

解 因为 s_0 是终结状态，且当输入是 0 时，有从 s_0 到自身的转移，故此机器识别所有零个或更多个连续的 0 组成的串。因为 s_4 也是终结状态，对于任何串，若以此串作为输入时从 s_0 所能达到的状态集中包含 s_4，则此串就能被识别。这样的串只有：零个或更多个连

续的 0 和后面跟 01 或 11 组成的串。因为 s_0 和 s_4 是仅有的终结状态，所以此机器识别的语言为 $\{0^n, 0^n01, 0^n11 \mid n \geq 0\}$。■

一个重要的事实是：非确定型有限状态自动机所能识别的语言也能被确定型有限状态自动机所识别。在下节中，我们将利用这个事实，以确定有限状态自动机能够识别哪些语言。

定理 1　如果语言 L 可以由非确定型有限状态自动机 M_0 所识别，则 L 也可以由一个确定型有限状态自动机 M_1 来识别。

证　我们来描述怎么从 M_0 构造识别 L 的确定型有限状态自动机 M_1。M_1 的每个状态都由 M_0 的状态集的一个子集构成，M_1 的初始符号是 $\{s_0\}$，即 M_0 的初始符号是 s_0 构成的集合。M_1 的输入集合与 M_0 的相同。对于 M_1 的一个给定状态 $\{s_{i_1}, s_{i_2}, \cdots, s_{i_k}\}$，输入串 x 将这个状态变为这个集合中元素的下一个状态构成的集合的并，即集合 $f(s_{i_1}, x), f(s_{i_2}, x), \cdots, f(s_{i_k}, x)$ 的并。M_1 的状态是 S 的所有子集，这里 S 是 M_0 的状态集。（如果此非确定型机器有 n 个状态，则确定型机器就有 2^n 个状态，因为所有子集都可以作为状态，包括空集，尽管实际使用的状态却很少。）如果 M_0 的一个子状态集含有终结状态，则它就为 M_1 的终结状态。

假设一个输入串可以由 M_0 识别，则用这个串从 s_0 出发可以达到的状态中有一个终结状态（读者应能对这一点作归纳证明）。这意味着：在 M_1 中，这个串能将 $\{s_0\}$ 引导至这样一个状态集：它是 M_0 的状态集的一个子集，且包含一个终结状态。这个子集是 M_1 的一个终结状态，所以这个串也能由 M_1 识别。还有，如果一个输入串不能由 M_0 识别，则它也就不能导致 M_0 中的任何终结状态。（读者应该能够给出它的详细证明。）因而这个输入串也不可能由 $\{s_0\}$ 导致 M_1 中的一个终结状态。□

例 9　求一个确定型有限状态自动机，能与例 7 中的非确定型有限状态自动机识别相同的语言。

解　所求的确定型有限状态自动机如图 10-13 所示，它是根据例 7 中的非确定型有限状态自动机构造的。此确定型自动机的状态是那个非确定型机器的状态集的子集。对于一个输入符号，一个子集的下一个状态是这样的集合：它由这个子集中的元素在那个非确定型机器中的所有下一个状态所构成。例如，对于输入 0，$\{s_0\}$ 转为 $\{s_0, s_2\}$，因为在那个非确定型机器中，s_0 有到它自己和 s_2 的转移；集合 $\{s_0, s_2\}$ 对输入 1 转为 $\{s_1, s_4\}$，因为在那个非确定型机器中，对输入 1 s_0 只转为 s_1，s_2 只转为 s_4；集合 $\{s_1, s_4\}$ 对输入 0 转为 $\{s_3\}$，因为在那个非确定型机器中，对输入 0 s_1 和 s_4 都只转为 s_3。所有以这种方法得到的子集都包括在这个确定型有限状态机器里。注意：空集也是这个机器的一个状态，因为它是 $\{s_3\}$ 对输入 1 的所有下一个状态构成的子集。初始状态是 $\{s_0\}$，终结状态是那些包含 s_0 或 s_4 的集合。■

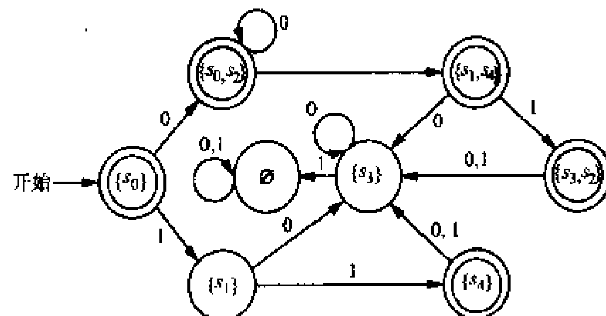


图 10-13　与例 7 中非确定型有限状态自动机等价的确定型有限状态自动机

练习

1. 设$A=\{0,11\}$，$B=\{00,01\}$，求下列集合：

   a) AB　　b) BA　　c) A^2　　d) B^3

2. 设 A 是串的一个集合。证明 $A\varnothing=\varnothing A=A$。
3. 求所有串的集合对 A 和 B 使得$AB=\{10,111,1010,1000,10111,101000\}$。
4. 证明下列等式成立：

   a) $\{\lambda\}^*=\{\lambda\}$

   b) 对任意串的集合 A，$(A^*)^*=A^*$

5. 对于下列集合 A，描述集合 A^*的元素：

   a) $\{10\}$　　b) $\{111\}$　　c) $\{0,\ 01\}$　　d) $\{1,\ 101\}$

6. 设 V 是一个字母表，A 和B 是 V^*的子集。证明：$|AB|\leqslant|A||B|$。
7. 设 V 是一个字母表，A 和 B 是 V^*的子集，且 $A\subseteq B$。证明：$A^*\subseteq B^*$。
8. 设 V 是一个字母表，A 是 V^*的子集。证明下列命题是正确的或是错误的：

   a) $A\subseteq A^2$　　b)如果 $A=A^2$，则 $\lambda\in A$　　c) $A\{\lambda\}=A$

   d) $(A^*)^*=A^*$　　e) $(A^*)A=A^*$　　f) $|A^n|=|A|^n$

9. 确定下列集合是否包含串 11101。

   a) $\{0,1\}^*$　　b) $\{1\}^*\{0\}^*\{1\}^*$　　c) $\{11\}\{1\}^*\{01\}$

   d) $\{11\}^*\{01\}^*$　　e) $\{111\}^*\{0\}^*\{1\}$　　f) $\{111,000\}\{00,01\}$

10. 确定下列串能否由图 10-9 中的确定型有限状态自动机所识别。

    a) 010　　b) 1101　　c) 111110　　d) 010101010

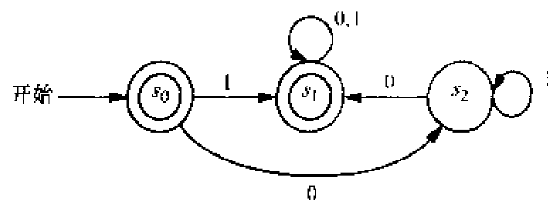
11. 对于下列每个集合，确定其中的每个串是否都能由图 10-9 中的确定型有限状态自动机所识别。

    a) $\{0\}^*$　　b) $\{0\}\{0\}^*$　　c) $\{1\}\{0\}^*$

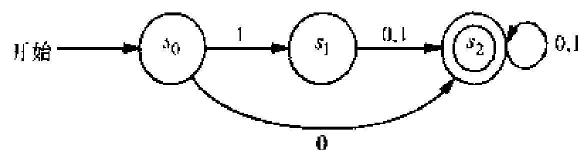
    d) $\{01\}^*$　　e) $\{0\}^*\{1\}^*$　　f) $\{1\}\{0,1\}^*$

在练习 12～16 中，求所给的确定型有限状态自动机所识别的语言。

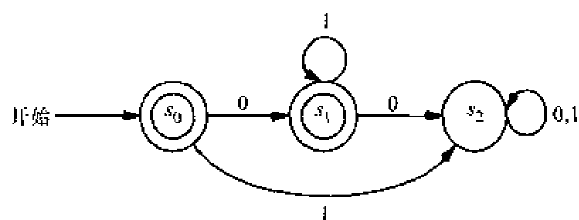
12.



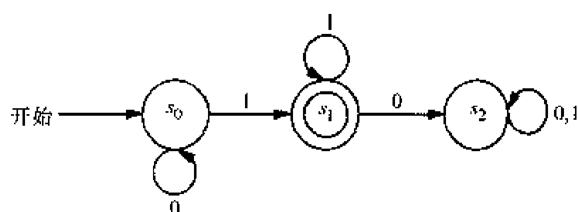
13.



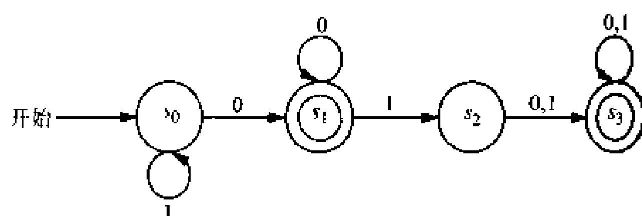
14.



15.

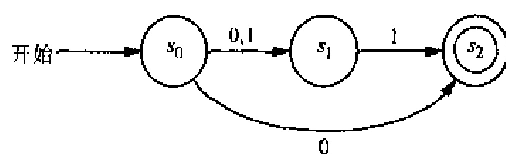


16.

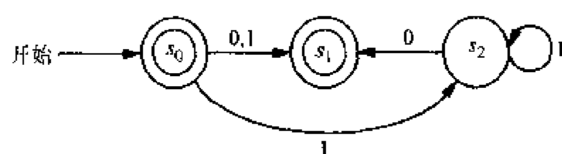


在练习 17~21 中，求所给的非确定型有限状态自动机所识别的语言。

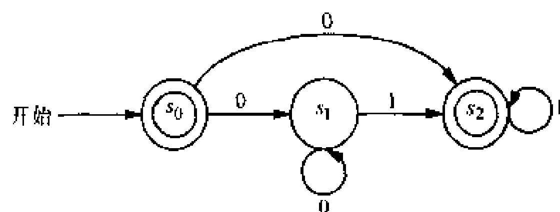
17.



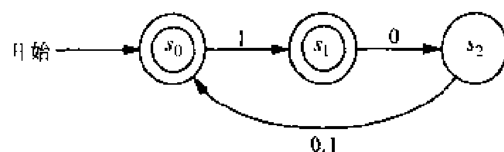
18.



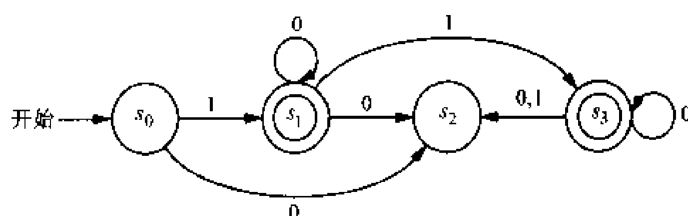
19.



20.



21.



22. 求一个确定型有限状态自动机，能与练习 17 中的非确定型有限状态自动机识别相同的语言。
23. 求一个确定型有限状态自动机，能与练习 18 中的非确定型有限状态自动机识别相同的语言。
24. 求一个确定型有限状态自动机，能与练习 19 中的非确定型有限状态自动机识别相同的语言。
25. 求一个确定型有限状态自动机，能与练习 20 中的非确定型有限状态自动机识别相同的语言。
26. 求一个确定型有限状态自动机，能与练习 21 中的非确定型有限状态自动机识别相同的语言。
27. 对于下列每个集合，分别求一个识别它的确定型有限状态自动机：
 a) $\{0\}$　　b) $\{1,00\}$　　c) $\{1^n \mid n=2,3,4,\cdots\}$
28. 对于练习 27 中的每种语言，分别求一个识别它的非确定型有限状态自动机，并且，如果可能的话，使之所具有的状态比你在练习 27 中所给的确定型有限状态自动机更少。

*29. 对于由个数相同的 0 和 1 组成的串构成的集合，证明没有有限状态自动机能够识别它。

10.4 语言的识别

10.4.1 引言

我们已经知道，有限状态自动机可以用作语言识别器。那么这些机器能识别什么样的集合呢？这个问题虽然看起来极为困难，但有限状态自动机所识别的语言还是有一个简单特征。这个问题由美国数学家克莱因（Stephen Kleene）于 1956 年首先解决。他证明了：一个集合能够被一个有限状态自动机识别当且仅当这个集合是这样得到的：以任意顺序通过对空集、空串和单字符串的连接、并或克莱因闭包构造出来。以这种方法构造出来的集合称为正则集合。

在 10.1 节中我们定义了正则文法。从名称上可以想到，有限状态自动机所识别的正则集合与正则文法间具有某种联系。特别是，一个集合是正则的当且仅当它可以由一个正则文法生成。

但也有一些集合不能由任何有限状态自动机识别，我们将给出这样一个集合。本节的最后还简短讨论了一些更强大的计算模型，如下推自动机和图灵机。

10.4.2 正则集合

正则集合是这样集合：从空集、空串、单字符集开始，以任意顺序通过连接、并和克莱

因闭包运算形成的。我们将知道，正则集合正是有限状态自动机识别的语言的集合。为定义正则集合，首先要定义正则表达式。

定义 1　集合 I 上的正则表达式的递归定义如下：

符号 $\varnothing$ 是一个正则表达式；

符号 λ 是一个正则表达式；

若 $x\in I$ 时，符号 x 是一个正则表达式；

当 **A**，**B** 是正则表达式时，符号 (**AB**)，(**A**∪**B** 和 $\mathbf{A}^*$ 都是正则表达式。

每个正则表达式都表示一个由下列规则指出的集合。

$\varnothing$ 表示空集，即没有串的集合；

λ 表示集合$\{\lambda\}$，即空串组成的集合；

x 表示集合$\{x\}$，它只包含单个符号 x 组成的串；

(**AB**)表示 **A** 和 **B** 代表的集合的连接；

(**A**∪**B**)表示 **A** 和 **B** 代表的集合的并；

$\mathbf{A}^*$ 表示 **A** 代表的集合的克莱因闭包。

正则表达式表示的集合称为*正则集合*。今后正则集合将由正则表达式来描述，所以，当我们提到正则集合 **A** 时，指的是此正则表达式 **A** 表示的集合。下面的例子说明了怎么用正则表达式来规定正则集合。

例 1　正则表达式 $\mathbf{10^*}$，$\mathbf{(10)^*}$，$\mathbf{0\cup 01}$，$\mathbf{0(0\cup 1)^*}$ 和 $\mathbf{(0^*1)^*}$ 所规定的正则集合中有哪些串？

解　这些正则表达式所表示的正则集合如表 10-9 所示。读者可以自己验证。■

表　10-9

表达式	串	表达式	串
$\mathbf{10^*}$	1 后面跟任意多个 0(也可以没有 0)	$\mathbf{0(0\cup 1)^*}$	以 0 开头的任意串
$\mathbf{(10)^*}$	10 的任意多次复制(包括空串)	$\mathbf{(0^*1)^*}$	不以 0 结尾的任意串
$\mathbf{0\cup 01}$	串 0 或 01		

10.4.3　克莱因定理

在 1956 年，克莱因证明了正则集合就是有限状态自动机识别的集合。因此，这个重要结论被称为克莱因定理。

定理 1　一个集合是正则的，当且仅当它可由一个有限状态自动机识别。

克莱因定理是自动机理论的中心定理之一，我们只证明必要性部分，即证明每个正则集合都可由一个有限状态自动机识别。充分性部分（有限状态自动机识别的集合都是正则的）留作练习。

证　回忆一下正则集合是通过正则表达式定义的，而正则表达式是递归定义的。所以，如果我们能证明下列事情，就证明了每个正则集合都可由一个有限状态自动机识别。

1. 证明 $\varnothing$ 可由一个有限状态自动机识别。
2. 证明$\{\lambda\}$可由一个有限状态自动机识别。

3. 证明$\{a\}$可由一个有限状态自动机识别，其中 a 是 I 中的符号。

4. 当 A 和 B 都可由有限状态自动机识别时，证明 AB 也可由有限状态自动机识别。

5. 当 A 和 B 都可由有限状态自动机识别时，证明 $A \cup B$ 也可由有限状态自动机识别。

6. 当 A 可由有限状态自动机识别时，证明 A^* 也可由有限状态自动机识别。 □

下面分别讨论每个任务。首先证明 $\emptyset$ 可以由有限状态自动机来识别。为此，需要构造一个没有终结状态的自动机。图 10-14a 就是这样一个自动机。

其次证明$\{\lambda\}$也可由有限状态自动机来识别。为此需要构造一个识别空串 λ 的自动机，且除 λ 外，它不识别任何其他的串。这个自动机可以这样来构造：初始状态 s_0 也用作终结状态，且对任何其他的串，没有转移能将 s_0 变为一个终结状态。图 10-14b 就是这样一个非确定型的自动机。

第三，证明$\{a\}$也可由非确定型的有限状态自动机来识别。为此构造如下机器：初始状态是 s_0，终结状态是 s_1，对于输入 a 有一个从 s_0 到 s_1 的转移，且没有其他转移。这个机器识别的唯一的串是 a。这个机器如图 10-14c 所示。

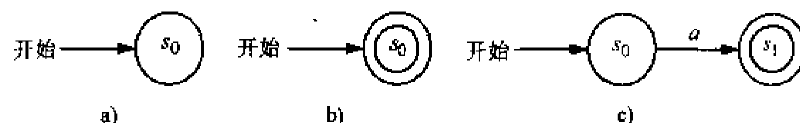


图 10-14 识别某些基本集合的非确定型有限状态自动机

下面证明：如果 A 和 B 都是可由有限状态自动机识别的语言，则 AB 和 $A \cup B$ 也可由有限状态自动机识别。设 A 是由$M_A = (S_A, I, f_A, s_A, F_A)$识别的，$B$ 是由 $M_B = (S_A, I, f_B, s_B, F_B)$识别的。

先来构造识别 A 与 B 的连接 AB 的有限状态自动机$M_{AB} = (S_{AB}, I, f_{AB}, s_{AB}, F_{AB})$，它是由识别 A 和 B 的两个机器串联而成的，使得 A 中的串将这个组合机器从 M_A 的初始状态 s_A 变为 M_B 的初始状态 s_B。B 中的串应该将这个组合机器从 s_B 变为此组合机器的一个终结状态。因此我们进行如下构造：令 S_{AB}是$S_A \cup S_B$；初始状态 s_{AB}就是s_A；终结状态集 F_{AB}或者是 M_B 的终结状态集，或者当且仅当 $\lambda \in A \cap B$ 时还包括 s_{AB}；M_{AB}的转移除了包括 M_A 和 M_B 中的全部转移之外，还有一些新的转移。对于 M_A 中每个导致终结状态的转移，在 M_{AB}中增加一个在同一个输入上从同一个状态到 s_B 的转移，这样，A 中的串就将 M_{AB}从 s_{AB}变为 s_B，然后 B 中的串将 s_B 变为 M_{AB}的一个终结状态。而且，对每个从 s_B 出发的转移，还在 M_{AB}中增加一个从 s_{AB}到同一个状态的转移。图 10-15a 包括了这个构造的所有说明。

现在构造识别 $A \cup B$ 的机器 $M_{A\cup B} = (S_{A\cup B}, I, f_{A\cup B}, s_{A\cup B}, F_{A\cup B})$。这个自动机是将 M_A 和 M_B 并行组合起来的，它使用一个新的初始状态，此初始状态具有 s_A 和 s_B 所具有的转移。令$S_{A\cup B} = S_A \cup S_B \cup \{s_{A\cup B}\}$，其中 $s_{A\cup B}$是$M_{A\cup B}$的新初始状态。当 $\lambda \in A \cup B$ 时，令终结状态集 $F_{A\cup B}$是$F_A \cup F_B \cup \{s_{A\cup B}\}$，否则为 $F_A \cup F_B$。$M_{A\cup B}$的转移除了包括M_A 和 M_B 中的所有转移外，还有如下的转移：对输入 i 从 s_A 到状态 s 的每个转移，包括一个对输入 i 从 $s_{A\cup B}$到状态 s 的转移；对输入 i 从 s_B 到状态 s 的每个转移，包括一个在输入 i 从 $s_{A\cup B}$到状态 s 的转移。这样，在这个新机器中，A 中的串将从 $s_{A\cup B}$导致一个终结状态，B 中的串也将从 $s_{A\cup B}$导致一个终结状态。图 10-15b 说明了 $M_{A\cup B}$的构造。

最后构造识别 A^* 的机器 $M_{A^*}=(S_{A^*},I,f_{A^*},s_{A^*},F_{A^*})$，其中 A^* 是 A 的克莱因闭包。令 S_{A^*} 包含 S_A 中所有状态，还有一个状态 s_{A^*}，它是这个新机器的初始状态。终结状态集 F_{A^*} 包含了 F_A 中所有状态和初始状态 s_{A^*}，因为必须要识别 λ。为了识别 A 中任意多个串的连接，我们包括 M_A 中的所有转移，以及如下转移：与从 s_A 出发的转移相匹配的从 s_{A^*} 出发的转移[⊖]，与从 s_A 出发的转移相匹配的从每个终结状态出发的转移。有了这个转移集，对于由 A 中的一些串连接而成的串，当 A 中的第一个串读完时，它将 s_A^* 变到一个终结状态，当 A 中的第二个串读完时，它又回到一个终结状态，等等。图 10-15c 说明了我们刚才的构造。

用以上描述的过程，可以对任意正则集合构造一个非确定型有限状态自动机。下面用例子来说明这一点。

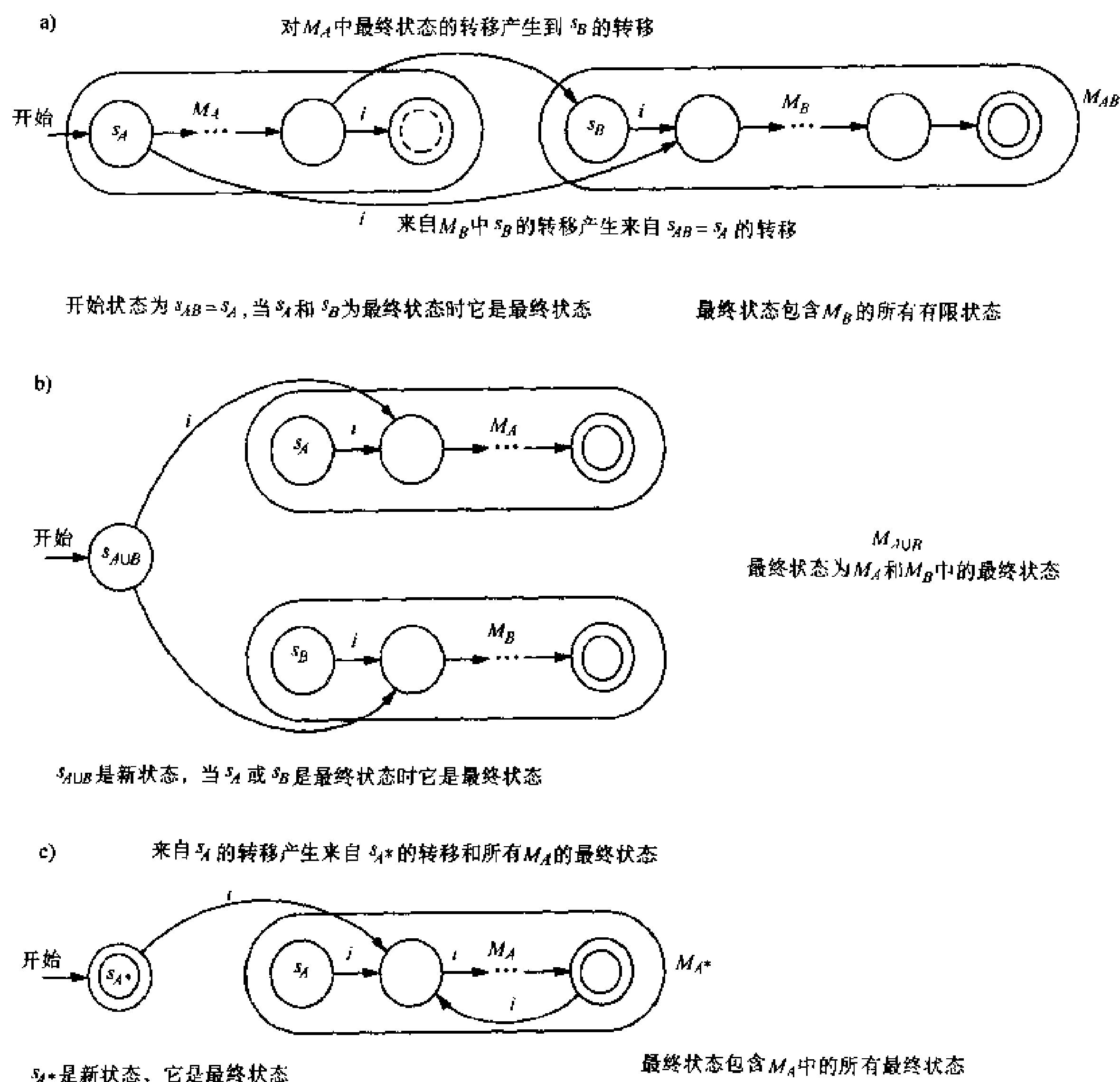


图 10-15　构造识别连接、并以及克莱因闭包的自动机

⊖ 即它们是在相同输入上到相同状态的转移。——译者注

例 2 构造一个非确定型有限状态自动机来识别正则集合 $1^* \cup 01$。

解 首先构造一个机器来识别 1^*。为此，先构造一个识别 1 的机器，再使用在定理 1 的证明中描述的构造 M_{A^*} 的方法。第二步，构造识别 **01** 的机器。先分别构造识别 **0** 和 **1** 的机器，再使用在定理 1 的证明中描述的构造 M_{AB} 的方法。最后，用在定理 1 的证明中描述的构造 $M_{A \cup B}$ 的文法，构造识别 $1^* \cup 01$ 的机器。所构造的有限状态自动机如图 10-16 所示。上述各个机器中的状态被标以不同的下标，即使对从前机器中继承下来的状态也是如此。注意：这样构造的机器并不是识别合 $1^* \cup 01$ 的最简单的机器。图 10-16b 是识别同一个集合但简单得多的机器。 ■

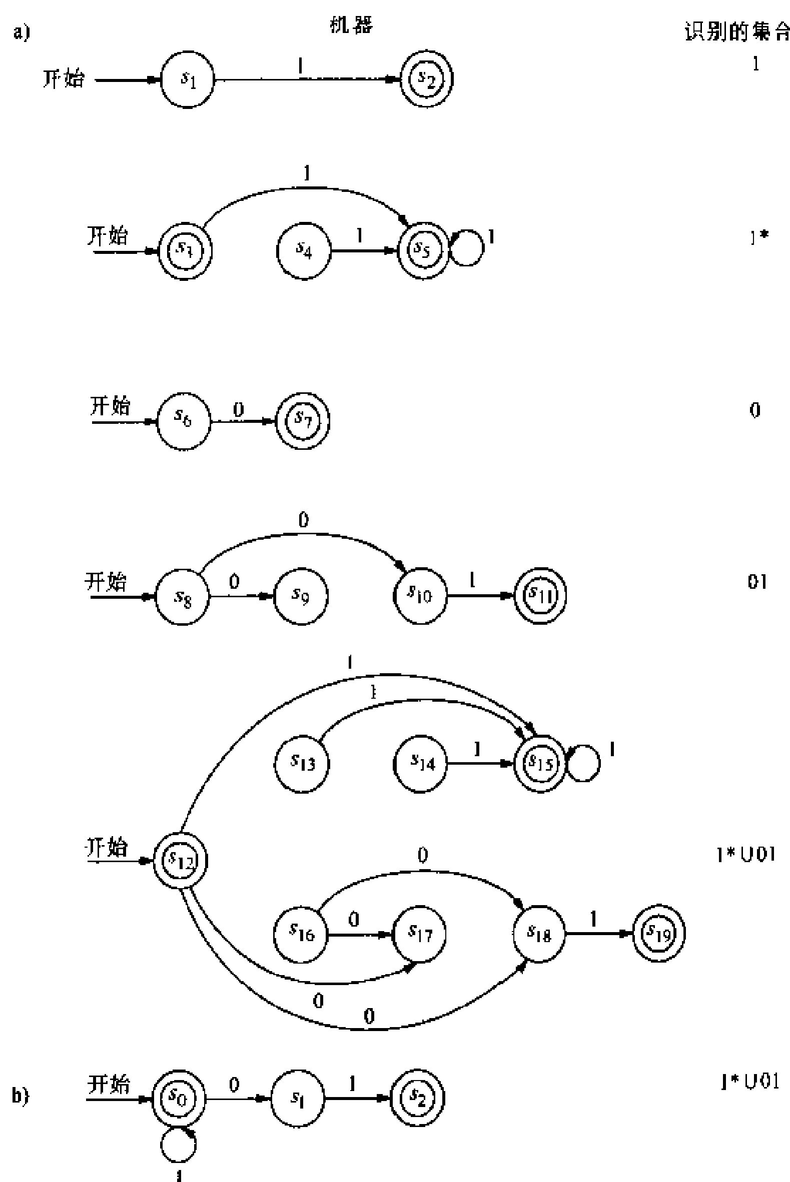


图 10-16 识别 $1^* \cup 01$ 的非确定型有限状态自动机

10.4.4 正则集合和正则文法

在 10.1 节中介绍了短语结构文法，还定义了各种不同类型的文法。特别定义了正则文

法（或 3 型文法），这是一个形如$G=(V,T,S,P)$的文法，文法的每个产生式的形式是 $S\to\lambda$，$A\to a$ 或$A\to aB$，其中 a 是终结符，A 和B 是非终结符。正如名称所暗示，正则文法和正则集合之间具有紧密的联系。

定理 2 一个集合可以由正则文法生成，当且仅当它是一个正则集合。

证 首先证明正则文法生成的集合是一个正则集合。设$G=(V,T,S,P)$是一个正则文法，其生成的集合是$L(G)$。为证明$L(G)$是正则的，我们只要构造一个非确定型有限状态自动机$M=(S,I,f,s_0,F)$来识别 $L(G)$。对 G 的每个非终结符A，状态集 S 都包含一个相应的状态，S 还包含一个状态s_F，它是一个终结状态。初始状态 s_0 是从初始符号 S 构造的。M 的转移是根据G 的产生式按以下方式构造的：如果 $A\to a$ 是一个产生式，则包括一个对输入 a 从s_A 到s_F 的转移；如果 $A\to aB$ 是一个产生式，则包括一个在输入 a 上从s_A 到s_B 的转移。终结状态集包括 s_F，如果 $S\to\lambda$ 是G 中产生式，则还要包括 s_0。不难证明，M 识别的语言与文法G 生成的语言相等，即$L(M)=L(G)$。这只要确定导致终结状态的词即可。详细证明留作练习。 □

在给出反方向的证明之前，先说明怎么构造一个非确定型机器，能识别由一个正则文法识别的集合。

例 3 构造一个非确定型有限状态自动机，使之识别正则文法$G=(V,T,S,P)$生成的语言，其中$V=\{0,1,A,S\}$，$T=\{0,1\}$，P 中的产生式为$S\to 1A$，$S\to 0$，$S\to\lambda$，$A\to 0A$，$A\to 1A$ 和$A\to 1$。

解 图 10-17 是识别$L(G)$的非确定型有限状态自动机的状态图。这个自动机是根据上面证明描述的过程构造的。在这个自动机中，s_0 是对应 S 的状态，s_1 是对应 A 的状态，s_2 是终结状态。 ■

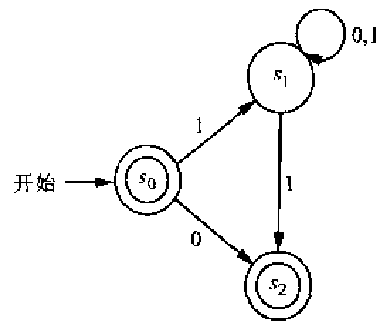


图 10-17 识别$L(G)$的非确定型有限状态自动机

现在来完成定理 2 的证明。

证 现在证明，如果一个集合是正则的，则存在一个正则文法生成它。设 M 是识别这个集合的一个有限状态机器，且具有性质：M 的初始状态s_0 对任何转移都不是下一个状态。(可以根据练习 14 找到这样机器。) 文法$G=(V,T,S,P)$的定义如下：G 之符号集 V 是这样形成的，对 S 的每个状态和I 中的每个输入符号，指派 V 中一个符号。G 的终结符集 T 是G 中这样符号构成的，它是根据 I 中输入符号构造的符号。初始符号 S 是根据初始状态 s_0 构造的符号。G 的产生式集P 是根据M 中的转移构造的。特别地，如果状态 s 对输入a 变到一个终结状态，则 P 中就包括产生式$A_s\to a$，其中 A_s 是根据状态s 构造的非终结符。如果状态 s 对输入a 变到状态t，则 P 中就包括产生式$A_s\to aA_t$。P 中包括产生式$S\to\lambda$，当且仅当$\lambda\in L(M)$。因为 G 的产生式对应于M 的转移，且导致终结符的产生式对应于导致终结状态的转移，因而不难证明$L(G)=L(M)$。详细证明留作练习。 □

下面的例子说明怎么根据自动机来构造文法，使得该文法生成的语言就是这个自动机识别的语言。

例 4 求一个正则文法，使之生成的集合是图 10-18 表示的有限状态自动机所识别的正

则集合。

解　生成该自动机所识别的集合的文法为 $G=(V,T,S,P)$，其中 $V=\{S,A,B,0,1\}$，其符号 S，A，B 分别对应于状态 s_0，s_1 和 s_2，$T=\{0,1\}$，S 是初始符号，产生式为 $S\to 0A$，$S\to 1B$，$S\to 1$，$S\to\lambda$，$A\to 0A$，$A\to 1B$，$A\to 1$，$B\to 0A$，$B\to 1B$ 和 $B\to 1$。■

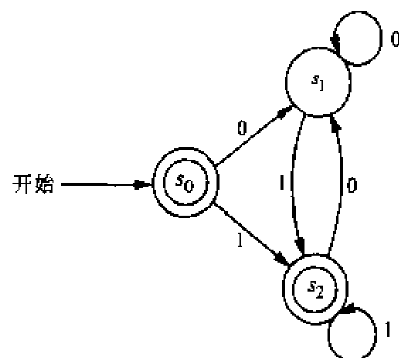


图 10-18　一个有限状态自动机

10.4.5　一个不能由有限状态自动机识别的语言

我们知道，一个集合能够由有限状态自动机识别当且仅当它是正则的。现在来证明存在不是正则的集合，方法是给出这样一个集合。证明这个集合不是正则的技术是一个重要方法，可以用来证明某类集合是不正则的。

例 5　集合 $\{0^n1^n \mid n=0,1,2,\cdots\}$ 是所有如下的串构成的：先是一列 0，后跟同样个数的 1。证明这个集合不是正则的。

解　假如这个集合是正则的，则存在一个有限状态自动机 $M=(S,I,f,s_0,F)$ 识别它。设 N 是这个机器中状态的个数，即 $N=|S|$。因为 M 能识别所有这样构成的串：先是一列 0，后跟同样个数的 1，故它必定能识别 0^N1^N。设 s_0，s_1，s_2，…，s_{2N} 是如下得到的状态序列：以 s_0 开始，以 0^N1^N 中的符号作为输入，且满足 $s_1=f(s_0,0)$，$s_2=f(s_1,0)$，…，$s_N=f(s_{N-1},0)$，$s_{N+1}=f(s_N,1)$，…，$s_{2N}=f(s_{2N-1},1)$。注意 s_{2N} 是一个终结状态。

因为只有 N 个状态，根据鸽巢原理，在 s_0，…，s_N 这头 $N+1$ 个状态中，至少有两个是相同的。假设 s_i，s_j 是两个这样相同的状态 $(0\leqslant i<j\leqslant N)$，则这表示 $f(s_i,0^t)=s_i$，其中 $t=i-j$。由此可知，t 次使用输入 0 后，可以得到一个从 s_i 回到它自己的循环，如状态图 10-19 所示。

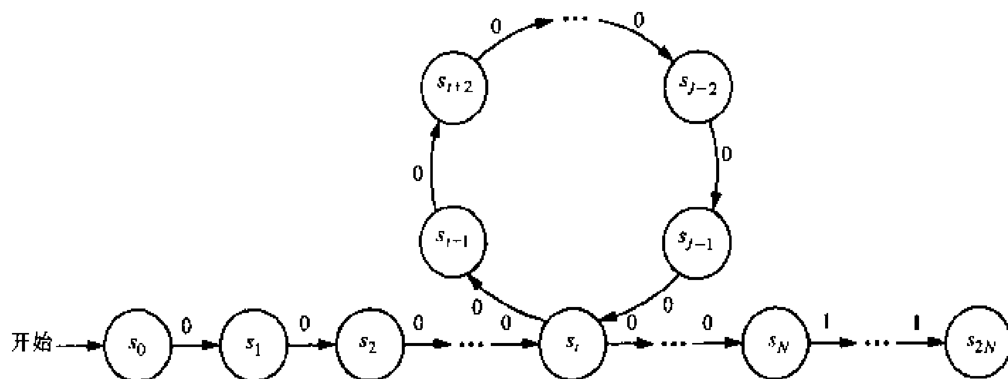


图 10-19　0^N1^N 所产生的路径

现在考虑输入串 $0^N0^t1^N=0^{N+t}1^N$。此串的前半部分比后半部分多了 t 个连续的 0。因为这个串不具有形式 0^n1^n（因为其中 0 的个数比 1 的个数多），故不能被 M 识别，因而 $f(s_0,0^{N+t}1^N)$ 也就不是终结状态。但当用 $0^{N+t}1^N$ 作为输入时，得到的结束状态与以前一样，即 s_{2N}。其理由是：此串中额外的 t 个 0 只是领着我们沿着那个循环多走一次，并将 s_i 再带回它自己（如图 10-19 所示）。然后，此串的剩余部分所导致的状态与以前完全相同。

这个矛盾证明了$\{0^n1^n \mid n=0,1,2,\cdots\}$不是正则的。∎

10.4.6 一些更强大的机器

很多计算都不能用有限状态自动机来完成，这类机器的局限性是它们只有有限的存储，因而限制了它们识别那些不是正则的语言，如$\{0^n1^n \mid n=0,1,2,\cdots\}$。因为一个集合是正则的当且仅当它是一个正则文法生成的语言，例 5 证明了没有正则文法能够生成集合$\{0^n1^n \mid n=0,1,2,\cdots\}$。但是，有一个上下文无关文法能够识别这个集合，此文法就是 10.1 节例 5 所给的文法。

由于有限状态自动机的局限性，有必要使用其他更加强大的计算模型。下推自动机就是这样一个模型。下推自动机除了包括有限状态自动机的所有部件外，还有一个栈，此栈能提供无限的存储。可以将符号放在栈顶上，也可从栈顶提走符号。下推自动机以两种方式识别集合。其一是，如果一个集合是所有这样的串构成的，当它们作为输入时产生空栈，则此集合能被识别。其二是，如果一个集合是所有这样的串构成的，当它们作为输入时导致终结状态，则此集合能被识别。可以证明，一个集合能被下推自动机识别，当且仅当它是一个上下文无关文法生成的语言。

但是，还有一些集合不能表示成上下文无关文法生成的语言，如集合$\{0^n1^n2^n \mid n=0,1,2,\cdots\}$。我们将指出为什么这个集合不能被下推自动机识别，但不给出证明，因为还没有介绍所需的方法。(但本章末尾的补充练习 28 给出了一个证明方法。) 可以使用栈来查看一个串是否以一列 0 开始，后再跟相同个数的 1，做法是对每个 0（只要仅读到多个 0 时）在栈上放一个符号，对每个 1（只要仅读到 0 后面的多个 1 时）从栈中去掉一个这样符号。但这个过程一旦完成，栈就空了，也就没法判断此串中是否还有与 0 个数相同的一列 2

还有一种比下推自动机更强大的机器，叫线性有界自动机，它能识别诸如$\{0^n1^n2^n \mid n=0,1,2,\cdots\}$的集合。特别地，线性有界自动机能够识别上下文相关文法。但是，这些机器不能识别短语结构文法生成的所有语言　为避免特殊类型机器的局限性，我们使用一种称为图灵机的模型，这种机器是以英国数学家图灵○命名的。图灵机除了包括有限状态自动机的所

○ 图灵（Alan Mathison Turing，1912—1954）图灵虽然是父亲在印度民政部供职时孕于母腹，但在伦敦出生。他在孩提时代就对化学和机械着迷，做过大量化学实验。图灵曾就读于英国的一所寄宿学校 Sherborne。1931 年，他获得了剑桥大学皇家学院的奖学金。在完成毕业论文后，他被选为该学院的成员。在毕业论文中，他重新发现了统计学中的一个著名定理——中心极限定理。1935 年，他对判定问题着了迷，这是伟大德国数学家希尔伯特提出的一个问题，问题是：是否有一个能用于判断任何命题是否为真的一般方法。图灵喜欢跑步（在他生命的后期，作为重要业余爱好，他经常参加比赛）。一天，在跑步之后的休息中，他发现了解决判定问题的关键思想。在他的解决方案中，他发明了现今称为图灵机的东西，并用它作为计算机器的最一般模型。利用这个机器，他发现了一个不能用一般方法判定的问题，这是一个关于他称为可计算数的问题。

从 1936 到 1938 年，图灵在普林斯顿大学访问，与丘奇（Alonzo Church）一起工作，丘奇也解决了希尔伯特的判定问题。在 1939 年，图灵回到了皇家学院。但在第二次世界大战爆发期间，他进入了英国外交部，从事对德国密码的分析工作。他对破解机械的德国密码机 Enigma 的密码作出了重要贡献，在赢得这次战争中起到了重要作用。

战后，图灵于从事早期计算机的开发。他对机器的思考能力产生了兴趣，他认为如果一台计算机在对问题的书面答复中与人没有区别，则应该认为它在"思考"。他还对生物学感兴趣，曾经写过关于有机体形式的形成和发展的书。1954 年，图灵服氰化物自杀，没有留下遗言作明确解释。也许，由于涉及同性恋关系的法律困扰和法院强迫他进行荷尔蒙治疗以减少性冲动，促使他决定结束自己的生命。

有部件外，还有一个带，其两端都是无限的。图灵机具有在带上读和写的能力，还能沿着带子左右移动。图灵机能够识别短语结构文法生成的所有语言。另外，它还能作为在计算机器上执行的所有计算的模型。由于这个能力，图灵机在理论计算机科学中得到了广泛得研究。下一节将作简要介绍。

练习

1. 用文字描述下列每个正则集合中的串。

   a) 1^*0　　b) 1^*00^*　　c) $111 \cup 001$

   d) $(1 \cup 00)^*$　　e) $(00^*1)^*$　　f) $(0 \cup 1)(0 \cup 1)^*00$

2. 判断 1011 是否属于下列正则集合。

   a) 10^*1^*　　b) $0^*(10 \cup 11)^*$　　c) $1(01)^*1^*$

   d) $1^*01(0 \cup 1)$　　e) $(10)^*(11)^*$　　f) $1(00)^*(11)^*$

   g) $(10)^*1011$　　h) $(1 \cup 00)(01 \cup 0)1^*$

3. 用正则表达式表达下列每个集合。

   a) 一个或更多的 0 后面跟一个 1 形成的串的集合。

   b) 两个或两个以上符号后面跟 3 个或 3 个以上 0 形成的串的集合。

   c) 一个 0 前没有 1 或一个 1 前没有 0 的串的集合。

   d) 集合包含这样的串：先是个数为 2 (mod 3) 的一串 1，后面是偶数个 0。

4. 构造确定型有限状态自动机来识别下列包含在 I^* 中的集合（其中 I 是一个字母表）。

   a) $\emptyset$　　b) $\{\lambda\}$　　c) $\{a\}$，其中 $a \in I$

*5. 若 A 是一个正则集合。证明：A 中串的反串构成的集合 A^R 也是正则的。

6. 求识别下列集合的有限状态自动机。

   a) $\{\lambda, 0\}$　　b) $\{0, 11\}$　　c) $\{0, 11, 000\}$

7. 用克莱因定理的证明中描述的构造方法，求识别下列集合的非确定型有限状态自动机。

   a) 0^*1^*　　b) $(0 \cup 11)^*$　　c) $01^* \cup 00^*1$

8. 构造非确定型有限状态自动机，用它识别正则文法 $G = (V, T, S, P)$ 生成的语言，其中：$V = \{0, 1, S, A, B\}$，$T = \{0, 1\}$，S 是初始符号，产生式集合为

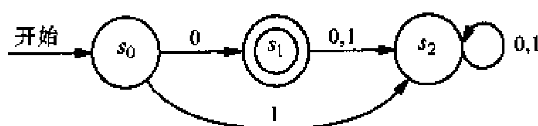
   a) $S \to 0A, S \to 1B, A \to 0, B \to 0$

   b) $S \to 1A, S \to 0, S \to \lambda, A \to 0B, B \to 1B, B \to 1$

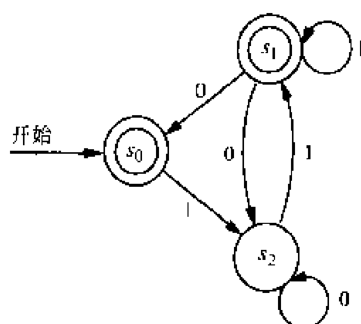
   c) $S \to 1B, S \to 0, A \to 1A, A \to 0B, A \to 1, A \to 0, B \to 1$

在练习 9~11 中，构造正则文法 $G = (V, T, S, P)$，使之生成的语言是所给的有限状态机识别的语言。

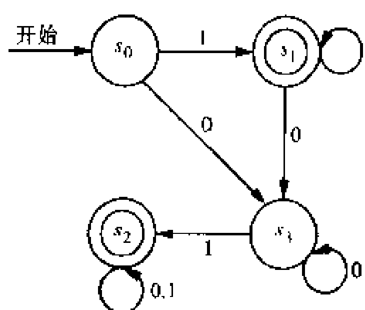
9.



10.



11.



12. 在定理 2 的证明中，从正则文法构造了一个有限状态自动机。证明：此自动机识别这个文法生成的集合。

13. 在定理 2 的证明中，从正则文法构造了一个有限状态自动机。证明：此文法生成这个自动机识别的集合。

14. 证明每个非确定型有限状态自动机等价于另一个这样的非确定型有限状态自动机，它的初始状态永不会被再次访问。

*15. 设 $M=(S,I,f,s_0,F)$ 是一个确定型有限状态自动机。证明：M 识别的语言 $L(M)$ 是无限的，当且仅当存在一个能被 M 识别的词 x 满足 $l(x)\geqslant|S|$。

*16. 用来证明某个集合不是正则的一个重要技术是**泵引理**。泵引理表述为：如果 $M=(S,I,f,s_0,F)$ 是一个确定型有限状态自动机，x 是 M 识别的语言 $L(M)$ 中的一个串，$l(x)\geqslant|S|$。则存在 I^* 中的串 u，v 和 w，使得 $x=uvw$，$l(uv)\leqslant|S|$，$l(v)\geqslant1$，且 $uv^iw\in L(M)(i=0,1,2,\cdots)$。证明泵引理。[提示：使用例 5 中的思想。]

*17. 证明集合 $\{0^{2n}1^n|n=0,1,2,\cdots\}$ 不是正则的。可以使用练习 16 中的泵引理。

*18. 证明集合 $\{1^{n^2}|n=0,1,2,\cdots\}$ 不是正则的。可以使用练习 16 中的泵引理。

*19. 证明 $\{0,1\}$ 上所有回文构成的集合不是正则的。可以使用练习 16 中的泵引理。[提示：考察形如 0^N10^N 的串。]

**20. 证明被有限状态自动机识别的集合是正则的。(这是克莱因定理的充分性部分。)

10.5 图灵机

10.5.1 引言

本章前面部分研究的有限状态自动机不能作为计算的一般模型，因为它们有其自身的局限性。例如，有限状态自动机虽然能识别正则集合，却不能识别许多很容易描述的集合，如 $\{0^n1^n|n\geqslant0\}$，计算机使用存储才能识别这些集合。可以用有限状态自动机来计算一些相对简单的函数（如两个数的和），但不能用它们来计算计算机所计算的函数（如两个数的积）。为克服这些不足，我们使用一种更强大的机器，称为图灵机，它是以著名数学家和计算机科学家图灵（Alan Turing）的名字命名的，他在 20 世纪 30 年代发明了这种机器。

图灵机主要由一个控制头和一个带组成，控制头在任何时候都处于有限多个不同状态中的某个状态，带子被分成许多方格，且两端都是无限的。当图灵机的控制头沿着带子来回移

动时，他能够在带上读和写，并根据所读的带符号改变状态。图灵机比有限状态自动机更强大，因为他有存储能力，而有限状态自动机却没有。我们将说明怎么用图灵机来识别集合，包括识别有限状态自动机不能识别的集合。还将说明怎么用图灵机来计算函数。图灵机是计算的最一般模型，本质上，它能做计算机能做的任何事情。

10.5.2 图灵机的定义

下面给出图灵机的形式定义。之后将根据它的控制头的动作来解释这个形式定义，控制头的动作包括读或写带上的符号及沿着带子左右移动。

定义 1 图灵机 $T=(S,I,f,s_0)$ 由下列各部分组成：有限状态集 S，包含空白符 B 的字母表 I，从 $S\times I$ 到 $S\times I\times\{R,L\}$ 的部分函数 f，及初始状态 s_0。

回忆 1.6 节练习 39 的引言，部分函数只对定义域中的某些元素有定义。这意味着上述部分函数 f 对于某些（状态，符号）对没有定义。但对于有定义的对，只有唯一一个三元组（状态，符号，方向）与之对应。

为用机器的观点来解释这个定义，考察控制头和带。如图 10-20 所示，带被分成小方格，且两端都是无限的，在任何时刻其上都只有有限多个非空白符。图灵机运行的每一步动作依赖于部分函数对当前状态和带符号的值。

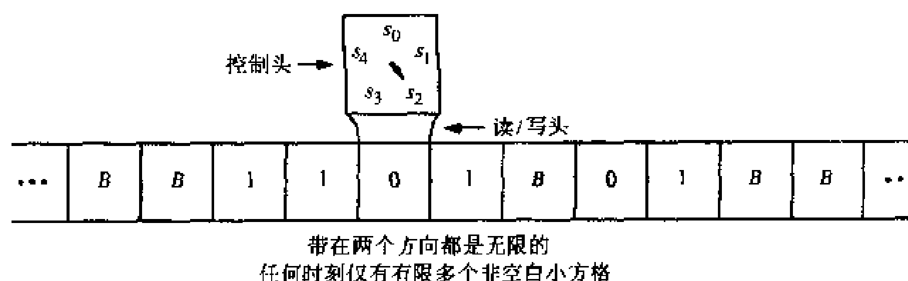


图 10-20 图灵机的表示

在每一步，控制头读的是当前带符号 x。如果控制头处于状态 s，且部分函数 f 在 (s,x) 处由 $f(s,x)=(s',x',d)$ 定义，则控制头

1）进入状态 s'。

2）在当前方格中擦掉 x，并写上符号 x'。

3）如果 $d=R$，向右移动一个方格；如果 $d=L$，向左移动一个方格。

我们将这一步写成五元组 (s,x,s',x',d)。如果部分函数 f 在 (s,x) 处没有定义，则图灵机 T 就停机。

定义一个图灵机的常用方法是指明形如 (s,x,s',x',d) 的五元组构成的一个集合。当使用这个定义时，就隐含地定义了状态集和输入字母表。

在运行开始的时候，总假设图灵机处于初始状态 s_0，且处于带中最左边的非空白符上。如果带中都是空白符，则控制头可以处于任何方格上。控制头所在的最左边非空白符位置称为该机器的*初始位置*。

下面的例子说明图灵机是怎么运行的。

例 1　下列 7 个五元组定义一个图灵机 T：$(s_0,0,s_0,0,R)$，$(s_0,1,s_1,1,R)$，(s_0,B,s_3,B,R)，$(s_1,0,s_0,0,R)$，$(s_1,1,s_2,0,L)$，(s_1,B,s_3,B,R)，$(s_2,1,s_3,0,R)$。当 T 在图 10-21a）所示的带上运行时，最后的带子是什么？

解　在开始运行时，T 处于状态 s_0，且在带中最左边的非空白符上。第一步，根据五元组$(s_0,0,s_0,0,R)$，读最左边的非空白方格中的 0，保持状态 s_0，在此方格中写下 0，向右移动一个方格。第二步，根据五元组 $(s_0, 1, s_1, 1, R)$，读当前方格中的 1，进入状态 s_1，在这方格中写下 1，向右移动一个方格。第三步，根据五元组$(s_1,0,s_0,0,R)$，读当前方格中的 0，进入状态 s_0，在这方格中写下 0，向右移动一个方格。第四步，根据五元组$(s_0,1,s_1,1,R)$，读当前方格中的 1，进入状态 s_1，在这方格中写下 1，向右移动一个方格。第五步，根据五元组$(s_1,1,s_2,0,L)$，读当前方格中的 1，进入状态 s_2，在这方格中写下 0，向左移动一个方格。第六步，根据五元组$(s_2,1,s_3,0,R)$，读当前方格中的 1，进入状态 s_3，在这方格中写下 0，向右移动一个方格。最后，机器在第七步停机，因为在这个机器的描述中，没有五元组是以$(s_3,0)$开头的。所有这些步骤如图 10-21 所示。

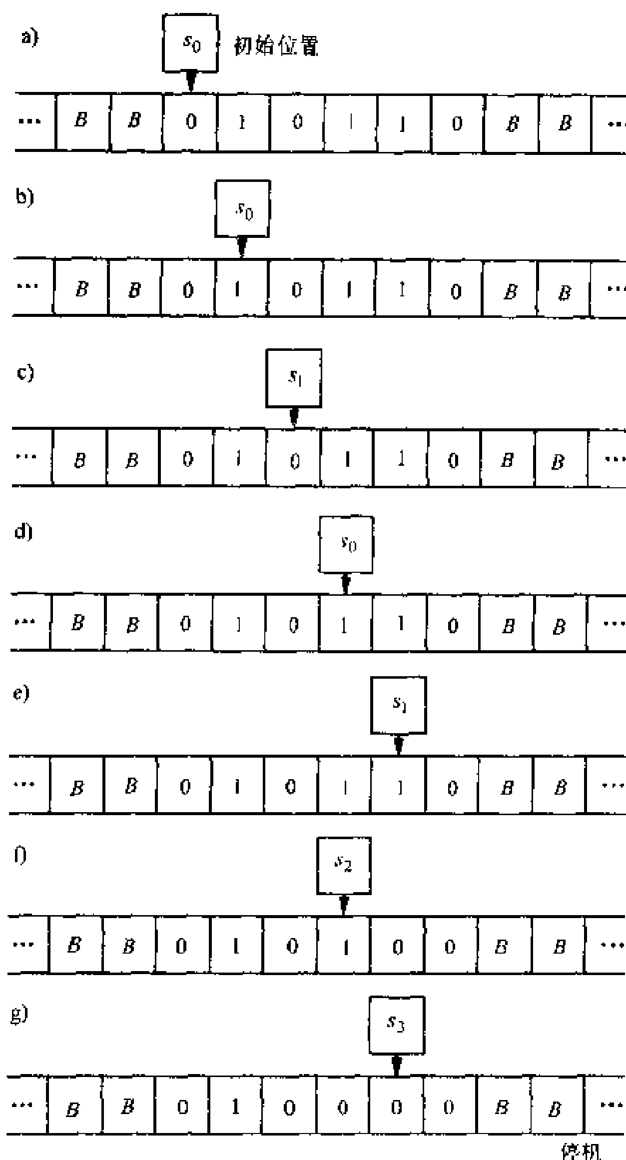


图 10-21　图灵机 T 在图 10-20 所示带上的运行步骤

注意，T 将带上第一对连续的 1 变为 0 后停机。■

10.5.3　用图灵机识别集合

可以用图灵机来识别集合。为此，如下定义终结状态的概念。图灵机 T 的终结状态是这样的状态：在描述 T 的五元组中，此状态不是任何五元组的第一个状态（例如，例 1 中的状态 s_3）。

现在定义图灵机识别一个串的含义是什么。给定一个串，我们在连续的方格中写下此串中的连续的符号。

定义 2　设 V 是字母表 I 的一个子集。图灵机 $T=(S,I,f,s_0)$识别 V^* 中串 x 当且仅当：若将 x 写在带上，T 从初始位置开始运行，则 T 能在一个终结状态停机。称 T 能识别

V^* 的子集 A，如果 x 能被 T 识别当且仅当 x 属于 A。

注意，为了识别 V^* 的子集 A，我们可以使用不在 V 中的符号。也就是说，输入字母表 I 也许包含不是 V 中的符号。这些额外的符号常用来当作标记（见例 3）。

什么情况下图灵机 T 不识别 V^* 中串 x 呢？答案是：设 x 的符号被放在 T 的带上的连续方格中，T 从初始位置开始运行，若 T 不停机，或者虽然停机，但不在终结状态停机，则 T 不识别 x。（读者应该明白，这是定义图灵机如何识别集合的许多方法中的一种。）

下面的例子说明了这个概念。

例 2　求一个图灵机，使之能识别第二位是 1 的二进制串构成的集合（即正则集合 $(\mathbf{0}\cup\mathbf{1})\mathbf{1}(\mathbf{0}\cup\mathbf{1})^*$）。

解　我们想要如下的图灵机：它从最左边的非空白带方格开始运行，然后向右移动，并且判断第二个符号是否为 1。若第二个符号是 1，则机器应该进入终结状态；如果第二个不是 1，则机器不能停机，或者在一个非终结状态下停机。

为构造这样的图灵机，应该包括五元组 $(s_0, 0, s_1, 0, R)$ 和 $(s_0, 1, s_1, 1, R)$ 来读第一个符号，并进入状态 s_1。下一步，添加五元组 $(s_1, 0, s_2, 0, R)$ 和 $(s_1, 1, s_3, 1, R)$ 来读第二个符号，且当这个符号是 0 时，进入状态 s_2。当这个符号是 1 时，进入状态 s_3。我们不希望使第二位是 0 的串也被识别，所以 s_2 不能是终结状态。我们希望 s_3 是终结状态。所以我们要加入五元组 $(s_2, 0, s_2, 0, R)$。因为我们不想识别空串和只有一位的串，所以还加入五元组 $(s_0, B, s_2, 0, R)$ 和 $(s_1, B, s_2, 0, R)$。

由上述 7 个五元组组成的图灵机 T 在终结状态 s_3 终止当且仅当：此二进制输入串至少有 2 位，并且第二位是 1。如果此二进制串少于两位，或者其第二位不是 1，则机器将在非终结状态 s_2 终止。■

对于一个正则集合，可以构造一个总是向右移动的图灵机来识别它（如例 2）。为构造这样的图灵机，先构造一个识别此集合的有限状态自动机，然后再用此有限状态自动机的转移函数构造一个图灵机，使之总向右移动。

下面说明怎么构造图灵机来识别非正则集合。

例 3　求识别集合 $\{0^n1^n \mid n \geq 1\}$ 的图灵机。

解　为构造这样图灵机，我们使用辅助带符号 M 作为标记。令 $V = \{0, 1\}$，$I = \{0, 1, M, B\}$。我们希望只识别 V^* 中的串。我们还有一个终结状态 s_6。图灵机依次用 M 替换串中最左边的 0，和用 M 替换串中最右边的 1，这样在带上左右移动。它能在一个终结状态终止当且仅当：这个串的构成是一列 0 后跟一列相同个数的 1。

虽然这很容易描述，图灵机也很容易执行，但我们想要使用的图灵机本身却有点复杂。标记 M 是用来跟踪已经检查过的最左边和最右边的符号。所用的五元组是：$(s_0, 0, s_1, M, R)$，$(s_1, 0, s_1, 0, R)$，$(s_1, 1, s_1, 1, R)$，(s_1, M, s_2, M, L)，(s_1, B, s_2, B, L)，$(s_2, 1, s_3, M, L)$，$(s_3, 1, s_3, 1, L)$，$(s_3, 0, s_4, 0, L)$，(s_3, M, s_5, M, R)，$(s_4, 0, s_4, 0, L)$，(s_4, M, s_0, M, R)，(s_5, M, s_6, M, R)。例如，当机器从开始一直运行到停机时，串 000111 将依次变成 $M00111$，$M0011M$，$MM011M$，$MM01MM$，$MMM1MM$，$MMMMMM$。（注意，此串不是根据

图灵机的操作步骤变化的。)

解释这个图灵机的动作和它为什么能识别集合$\{0^n1^n | n \geq 1\}$将留给读者作为练习（本节末的练习 13)。 ■

可以证明：一个集合能被图灵机识别当且仅当它是 0 型文法生成的集合，即短语结构文法生成的集合。这里略去它的证明。

10.5.4 用图灵机计算函数

图灵机可以看作是能求部分函数的值的计算机。为理解这一点，假设给定输入串 x 时图灵机 T 能够停机，且停机时，串 y 在它的带上。因此可以定义 $T(x)=y$。T 的定义域是使 T 能停机的串构成的集合。对于输入 x，若 T 不停机，则 T（x）没有定义。将图灵机看成计算串的函数值的机器是有用的，但怎么用图灵机来计算定义在整数、整数对或整数三元组等上的函数呢?

为将图灵机看作是计算 k 元非负整数集合到非负整数集合的函数（这样的函数称为*数论函数*）的计算机，需要找到在带上表示 k 元整数的方法。为此，我们使用整数的一元表示，即将非负整数 n 表示成有 $n+1$ 个 1 的串。例如，0 被表示成串 1，5 被表示成串 111111。为表示 k 元组$(n_1,n_2,\cdots,n_k)$，我们先写 n_1+1 个 1，后面跟一个星号，再跟 n_2+1 个 1，再跟一个星号，等等，以 n_k+1 个 1 结尾。例如，四元组(2,0,1,3)可以表示成串 111 * 1 * 11 * 1111。

现在能将图灵机看成计算一系列数论函数 $T,T^2,\cdots,T^k,\cdots$。函数 T^k 是根据 T 在 k 元整数组上的动作定义的，当然，k 元整数组被表示成用星号隔开的一些一元表示。

例 4 构造一个图灵机将两个非负整数相加。

解 需要构造图灵机来计算函数$f(n_1,n_2)=n_1+n_2$。元素对(n_1,n_2)被表示成由这样的串：先是 n_1+1 个 1，后面跟一个星号，再跟 n_2+1 个 1。机器 T 应以这个串作为输入，并在带上产生 n_1+n_2+1 个 1 作为输出。实现这个任务的一个方法如下：机器从输入串最左边的 1 开始运行，执行去掉这个 1 的步骤。若 $n_1=0$，则停机，此时，星号之前已没有 1 了。在剩下的 1 中，以最左边的 1 替换星号，然后停机。下列五元组能作到这一点：$(s_0,1,s_1,B,R)$，$(s_1,*,s_3,B,R)$，$(s_1,1,s_2,B,R)$，$(s_2,1,s_2,1,R)$，$(s_2,*,s_3,1,R)$。 ■

不幸的是，即使是相对简单的函数，要构造图灵机来计算它也是极为费力的。例如，在许多书中都有计算两个非负整数乘积的图灵机，此图灵机有 31 个五元组和 11 个状态。如果构造计算相对简单的函数的图灵机都是挑战性的，那么我们对构造更加复杂函数的图灵机还有什么指望呢？简化这个问题的一个方法是使用多带图灵机（它同时使用不止一个带子)，并给出构造复合函数的多带图灵机的方法。可以证明：对任何多带图灵机，存在一个单带图灵机，使得他们能做完全相同的事情。

可由图灵机计算的函数称为*可计算的*。可以很直接地证明：存在不是可计算的数论函数，但要实际给出这样一个函数却不容易。在本节末尾的练习 23 中，其引言定义了一个*忙碌海狸函数*，这是一个不可计算函数。证明忙碌海狸函数是不可计算的一种方法是证明它比任何可计算函数都增长得快（见练习 24)。

10.5.5 不同类型的图灵机

图灵机的定义有许多变种。可用很多方法来扩展图灵机的能力。例如，可以允许图灵机在一步中左移、右移或根本不动；允许图灵机操作多个带子，n 个带的图灵机可以用$(2+3n)$元组来描述；允许带是二维的，即在每一步可以上下左右移动，而不像在一维带上那样只向左或右移动；还可以允许有多个带头，它们能同时读不同的方格。还有，可以允许图灵机是非确定的，即允许（状态，带符号）对作为头两个元素出现在图灵机的多个五元组中。也可以用多种方法来削减图灵机的能力。例如，可以限制带只在一个方向是无限的；可以限制带字母表只有两个符号。图灵机的所有这些变种都已被详细地研究。

重要的是：不管使用哪个变种图灵机，或使用变种图灵机的哪个组合，都决不会增加或减少机器的能力。这些变种的任何一个能做的事，本节定义的图灵机都能做到，反之亦然。这些变种之所以还有用，是因为：有些时候，在做某些特殊任务时，使用它们比只使用定义 1 定义的图灵机容易得多。它们永远不会扩展机器的能力。

10.5.6 丘奇-图灵论题

图灵机还是相对简单的。它只能有有限多个状态，每一次它们只能在一维带上读或写一个符号。但结果表明，图灵机是极其强大的，我们已经看到，可以构造图灵机来执行数的加法和乘法。对于算法所计算的特殊函数，虽然很难实际构造图灵机来计算它们，但这样的图灵机总是能够找到的。这也正是图灵发现这种机器的目的。

*丘奇-图灵论题*是说：对于任何可用有效算法来解的问题，都存在解该问题的图灵机。可以用大量的证据来说明丘奇-图灵论题，但它还是被称为是*论题*，而不是定理，这是因为：有效算法说明的*可解性*概念是非形式的、且不严格的，相反，图灵机定义的可解性概念是形式的、且是严格的。当然，对于任何问题，只要它能够用装备了某个语言写成的程序的计算机来解，即使使用了无限多的存储，都应该被认为是有效可解的[㊀]。

人们发明了许多形式理论来刻画有效可计算性概念，其中有图灵的理论、丘奇[㊁]的 λ 演算以及克莱因和波斯特提出的理论。这些理论表面上看起来十分不同，但令人惊奇的是，它们都是等价的，因为可以证明它们定义了完全相同的函数类。由此证据可以看出，图灵的思想虽然是在现代计算机的发明之前形成的，但确实描述了计算机最根本的能力。

练习

1. 设 T 是下列五元组定义的图灵机：$(s_0,0,s_1,1,R)$，$(s_0,1,s_1,0,R)$，$(s_0,B,s_1,0,R)$，$(s_1,0,s_2,1,L)$，$(s_1,1,s_1,0,R)$，$(s_1,B,s_2,0,L)$。对于下列初始带，判断 T 停机时的最终带。假设 T 从初始位置开始执行。

㊀ “有效可解”在有些参考书中又称为“能行可解”。——译者注

㊁ 丘奇（Alonozo Church，1903—1995） 丘奇出生于华盛顿特区，曾在哥廷根跟随希尔伯特学习，后来转到阿姆斯特丹。从 1927 年到 1967 年，他执教于普林斯顿大学，1967 年调到加州大学洛杉矶分校（UCLA）。丘奇是符号逻辑学会（Association of Symbolic Logic）的创始人之一。他对可计算性理论作出了实质性的贡献，其中包括对判定问题的解、λ 演算的发明，以及对现今称为丘奇-图灵论题的陈述。克莱因和图灵都是丘奇的学生。他在九十岁生日后还在发表文章。

a)

…	B	B	0	0	1	1	B	B	…

b)

…	B	B	1	0	1	B	B	B	…

c)

…	B	B	1	1	B	0	1	B	…

d)

…	B	B	B	B	B	B	B	B	…

2. 设 T 是下列五元组定义的图灵机：$(s_0, 0, s_1, 0, R)$，$(s_0, 1, s_1, 0, L)$，$(s_0, B, s_1, 1, R)$，$(s_1, 0, s_2, 1, R)$，$(s_1, 1, s_1, 1, R)$，$(s_1, B, s_2, 0, R)$，$(s_2, B, s_3, 0, R)$。对于下列初始带，判断 T 停机时的最终带。假设 T 从初始位置开始执行。

a)

…	B	B	0	1	0	1	B	B	…

b)

…	B	B	1	1	1	B	B	B	…

c)

…	B	B	0	0	B	0	0	B	…

d)

…	B	B	B	B	B	B	B	B	…

3. 对于由五元组$(s_0,0,s_0,0,R)$,$(s_0,1,s_1,0,R)$,(s_0,B,s_2B,R),$(s_1,0,s_1,0,R)$,$(s_1,1,s_0,1,R)$和(s_1,B,s_2,B,R)描述的图灵机,给定一个二进制串输入,它能做什么?
4. 对于由五元组$(s_0,0,s_1,B,R)$,$(s_0,1,s_1,1,R)$,$(s_1,0,s_1,0,R)$,$(s_1,1,s_2,1,R)$,$(s_2,0,s_1,0,R)$,$(s_2,1,s_3,0,L)$,$(s_3,0,s_4,0,R)$和$(s_3,1,s_4,0,R)$描述的图灵机，给定一个二进制串输入，它能做什么?
5. 构造一个带符号为 0、1 和 B 的图灵机,它将带上第一个 0 替换为 1,而其余符号保持不变。
6. 构造一个带符号为 0、1 和 B 的图灵机，对于给定的二进制输入串，它将带上所有 0 替换为 1，而所有的 1 保持不变。
7. 构造一个带符号为 0、1 和 B 的图灵机，对于给定的二进制输入串，它将带上最左边的 1 以外的所有 1 替换为 0，而其余符号保持不变。
8. 构造一个带符号为 0、1 和 B 的图灵机，对于给定的二进制输入串，它将带上首先出现的两个连续的 1 替换为 0，而其余符号保持不变。
9. 构造一个图灵机，它识别的集合是所有以 0 结尾的二进制串组成的集合。
10. 构造一个图灵机，它识别的集合是所有至少包含两个 1 的二进制串组成的集合。
11. 构造一个图灵机，它识别的集合是所有包含偶数个 1 的二进制串组成的集合。
12. 对于例 3 中的图灵机，若从下列每个串开始运行，写出其每一步的带内容。

    a) 0011　　b) 00011　　c) 101100　　d) 000111
13. 例 3 中的图灵机识别一个串当且仅当此串具有形式 0^n1^n（其中 n 是一个正整数），试说明原因。

*14. 构造识别集合$\{0^{2n}1^n \mid n \geq 0\}$的图灵机。

*15. 构造识别集合$\{0^n1^n2^n \mid n \geq 0\}$的图灵机。

16. 构造一个图灵机计算函数$f(n) = n + 2$，其中 n 是非负整数。
17. 构造一个图灵机计算下列函数：当 $n \geq 3$ 时，$f(n) = n - 3$，当 $n = 0,1,2$ 时，$f(n) = 0$，其中 n 是非负整数。
18. 构造一个图灵机计算函数$f(n) = n \bmod 3$。

19. 构造一个图灵机计算下列函数：当 $n \geq 5$ 时，$f(n)=3$；当 $n=0, 1, 2, 3$ 或 4 时，$f(n)=0$。
20. 构造一个图灵机计算下列函数：对于所有非负整数 n_1 和 n_2，$f(n_1,n_2)=n_2+2$。
*21. 构造一个图灵机计算下列函数：对于所有非负整数 n_1 和 n_2，$f(n_1,n_2)=\min\{n_1,n_2\}$
22. 构造一个图灵机计算下列函数：对于所有非负整数 n_1 和 n_2，$f(n_1,n_2)=n_1+n_2+1$。

设 $B(n)$ 是具有 n 个状态且字母表为 $\{1,B\}$ 的图灵机从空白带开始运行后在带上所能打印的 1 的最大个数。根据给定的值 n 确定 $B(n)$ 这个问题称为*忙碌海狸问题*，该问题由拉多（Tibor Rado）于 1962 年首先研究。现在已经知道：$B(2)=4, B(3)=6, B(4)=13$。但当 $n \geq 5$ 时，$B(n)$ 等于什么还不知道。

*23. 通过寻找下面的图灵机来证明 $B(2)$ 至少是 4：该图灵机有两个状态，字母表是 $\{1,B\}$，且在停机时，带上有 4 个连续的 1。
**24. 证明函数 $B(n)$ 不能用任何图灵机来计算。[提示：假设有一个图灵机计算二进制表示的 $B(n)$，构造一个图灵机 T，从空带开始，写下 n 的二进制表示，计算 $B(n)$ 并表示成二进制数，然后将 $B(n)$ 从二进制表示转换为一元表示。再证明：当 n 充分大时，T 的状态个数可以小于 $B(n)$，导致矛盾。]

关键术语和结果

术语

字母表（或词汇表）：用来构造串的元素组成的集合

语言：字母表上所有串构成的集合的一个子集

短语结构文法（V, T, S, P）：语言的一种描述，包括字母表 V、终结符集 T、初始符号 S 和产生式集 P

产生式 $w \to w_1$：只要语言的某个串中出现了 w，就可将此串中的 w 替换为 w_1。

$w_1 \Rightarrow w_2$（*由 w_1 可直接派生 w_2*）：w_2 是从 w_1 按如下方式得到的：用产生式将 w_1 中的某个串替换为另一个串

$w_1 \stackrel{*}{\Rightarrow} w_2$（*由 w_1 可直接派生 w_2*）：w_2 是从 w_1 按如下方式得到的：用一系列产生式将某些串替换为另一些串

0 型文法：任意短语结构文法

1 型文法：是一种短语结构文法，但其产生式都具有形式 $w_1 \to w_2$，其中 $l(w_1) \leq l(w_2)$ 或 $w_2=\lambda$

2 型（或上下文无关）文法：是一种短语结构文法，但其产生式都具有形式 $A \to w_1$，其中 A 是一个非终结符

3 型（或正则）文法：是一种短语结构文法，但其产生式的形式是 $A \to aB$, $A \to a$ 或 $S \to \lambda$，其中 A, B 是非终结符，S 是初始符，a 是一个终结符

派生（或语法分析）树：一个带根的有序树，其根表示 2 型文法的初始符，内部结点表示非终结符，叶表示终结符，结点的儿子是产生式右边的符号，按从左到右顺序排列，同时，父亲表示的符号都在左边

巴科斯–诺尔范式：上下文无关文法的一种描述，在这种描述中，左边非终结符相同的所有产

生式被合并成一个式子，式子的右边是这些产生式不同的右边，并用竖线符将其分开，用尖括号将非终结符括起来，符号→被换成∷=

有限状态机器(S,I,O,f,g,s_0)(或米利机)：一个六元组，包括状态集S，输入字母表I，输出字母表O，转移函数f(对每个状态与输入对，指派下一个状态)，输出函数g(对每个状态与输入对，指派一个输出)和一个初始符s_0

AB(A和B的连接)：由A中串和B中串连接而成的串构成的集合

A^*(A的克莱因闭包)：由A中任意多个串连接而成的串构成的集合

确定型有限状态自动机(S,I,f,s_0,F)：一个五元组，包括状态集S，输入字母表I，转移函数f(对每个状态与输入对，指派下一个状态)，初始符s_0和终结状态集F

非确定型有限状态自动机(S,I,f,s_0,F)：一个五元组，包括状态集S，输入字母表I，转移函数f(对每个状态与输入对，指派下一个可能状态的集合)，初始符s_0和终结状态集F

自动机识别的语言：将自动机从初始状态带到终结状态的输入串构成的集合。

正则表达式：如下递归定义的表达式：$\emptyset$，λ和每个输入符号x都是正则表达式；当A和B是正则表达式时，(AB)，(AB)和A^*都是正则表达式

正则集合：正则表达式定义的集合

图灵机$T=(S,I,f,s_0)$：由下列各部分组成的四元组：有限状态集S，包含空白符B的字母表I，从$S\times I$到$S\times I\times\{R,L\}$的一个部分函数，初始状态s_0

结果

对每个非确定型有限状态自动机，存在一个确定型有限状态自动机，它们识别相同的集合。

克莱因定理：一个集合是正则的，当且仅当它可由一个有限状态自动机来识别

一个集合是正则的当且仅当它可由一个正则文法生成。

复习题

1. a) 定义短语结构文法
   b) “一个串可以由短语结构文法从串w派生出来”的含义是什么？
2. a) 什么是短语结构文法生成的语言？
   b) 设短语结构文法G如下：词汇表为$\{S,0,1\}$，终结符集为$T=\{0,1\}$，初始符号为S，产生式为$S\to 000S$和$S\to 1$。G生成的语言是什么？
   c) 给出生成集合$\{01^n \mid n=0,1,2,\cdots\}$的短语结构文法。
3. a) 定义一个1型文法。
   b) 给出一个是文法却不是1型文法的例子。
   c) 定义2型文法。
   d) 给出一个是1型文法却不是2型文法的例子。
   e) 定义一个3型文法（或上下文无关文法）。
   f) 给出一个是2型文法却不是3型文法的例子。
4. a) 定义一个正则文法。
   b) 定义一个正则语言。
   c) 给出一个是3型文法却不是正则文法的例子
   d) 证明集合$\{0^m1^n \mid m、n=0,1,2,\cdots\}$是一个正则语言。
5. a) 什么是巴科斯-诺尔范式？

b）选择英语的一个子集，给出其巴科斯–诺尔范式。

6. a）什么是一个有限状态机？

b）怎么用有限状态机建立下列自动售货机的模型：它只接受 15 分硬币，在放入 75 分之后，它发售一筒软饮料。

7. a）什么是一个串集合的克莱因闭包？

b）求集合{11,0}的克莱因闭包。

8. a）定义一个有限状态自动机。

b）"一个串由一个有限状态自动机识别"的含义是什么？

9. a）定义一个非确定型有限状态自动机。

b）试证：对每个非确定型有限状态自动机，存在一个确定型有限状态自动机，它们识别相同的语言。

10. a）定义一个集合 I 上的正则表达式集。

b）解释怎么用正则表达式表示正则集合。

11. 叙述克莱因定理。

12. 试证：一个集合可由正则文法生成当且仅当它是一个正则集合。

13. 给出一个不能由有限状态自动机识别的集合的例子，并证明没有有限状态自动机能够识别它。

14. 定义一个图灵机。

15. 描述怎么用图灵机来识别集合。

16. 描述怎么用图灵机来计算数论函数。

补充练习

*1. 求识别下列每个语言的一个短语结构文法。

a）形如 $0^{2n}1^{3n}$ 的二进制数串的集合，其中 n 是一个非负整数。

b）二进制数串的集合：其中 0 的个数是 1 的个数的两倍。

c）形如 w^2 的二进制数串的集合，其中 w 是二进制数串。

*2. 求产生集合 $\{0^{2^n} \mid n \geq 0\}$ 的一个短语结构文法。

在练习 3 和 4 中，$G=(V,T,S,P)$是一个上下文无关文法，其中 $V=\{(,),S,A,B\}$，$T=\{(,)\}$，S 是初始符号，产生式有 $S\rightarrow A$，$A\rightarrow AB$，$A\rightarrow B$，$B\rightarrow (A)$，$B\rightarrow ()$，$S\rightarrow\lambda$。

3. 构造下列串的派生树。

a）(())　　b）()(())　　c）((()()))

*4. 证明 $L(G)$ 就是第 3 章中定义的括号的合式串集合。

称一个上下文无关无法是歧义的，如果 $L(G)$ 中有一个词有两个派生，且将这两个派生看作带根的有序树时，产生两个不同的派生树。

5. 设文法 $G=(V,T,S,P)$ 为：$V=\{0,S\}$，$T=\{0\}$，S 是初始符号，产生式有 $S\rightarrow 0S$，$S\rightarrow S0$ 和 $S\rightarrow 0$。构造 0^3 的两个不同派生树，从而证明 G 是歧义的。

6. 设文法 $G=(V,T,S,P)$ 为：$V=\{0,S\}$，$T=(0)$，S 是初始符号，产生式有 $S\rightarrow 0S$ 和 $S\rightarrow 0$。证明 G 是非歧义的。

7. 设 A 和 B 是 V^* 的两个有限子集，其中 V 是一个字母表。问 $|AB|=|BA|$ 肯定成立吗？

8. 设 V 是一个字母表，A，B 和 C 是 V^* 的子集。证明或反驳下列各式。

   a) $A(B\cup C)=AB\cup AC$

   b) $A(B\cap C)=AB\cap AC$

   c) $A(BC)=(AB)C$

   d) $(A\cup B)^*=A^*\cup B^*$

9. 设 V 是一个字母表，A 和 B 是 V^* 的子集。从 $A^*\subseteq B^*$ 能否推出 $A\subseteq B$？

10. 正则表达式 $(\mathbf{2}^*)(\mathbf{0}\cup(\mathbf{12}^*))^*$ 表示的串集合是什么（串的符号在集合 $\{0,1,2\}$ 中）？

如下递归定义集合 I 上正则表达式的星高度 $h(\mathbf{E})$：

$h(\emptyset)=0$；

若 $x\in I$，则 $h(x)=0$；

若 $\mathbf{E}_1,\mathbf{E}_2$ 是正则表达式，则 $h(\mathbf{E}_1\cup\mathbf{E}_2)=h((\mathbf{E}_1\mathbf{E}_2))=\max\{h(\mathbf{E}_1),h(\mathbf{E}_2)\}$；

若 $\mathbf{E}$ 是正则表达式，则 $h(\mathbf{E}^*)=h(\mathbf{E})+1$。

11. 求下列正则表达式的星高度：

   a) $\mathbf{0}^*\mathbf{1}$　　b) $\mathbf{0}^*\mathbf{1}^*$

   c) $(\mathbf{0}^*\mathbf{01})^*$　　d) $((\mathbf{0}^*\mathbf{1})^*)^*$

   e) $(\mathbf{010}^*)(\mathbf{1}^*\mathbf{01}^*)^*((\mathbf{01})^*(\mathbf{10})^*)^*$　　f) $(((((\mathbf{0}^*)\mathbf{1})^*\mathbf{0})^*)\mathbf{1})^*$

*12. 对下列每个正则表达式，求一个表示相同语言但具有最小星高度的正则表达式。

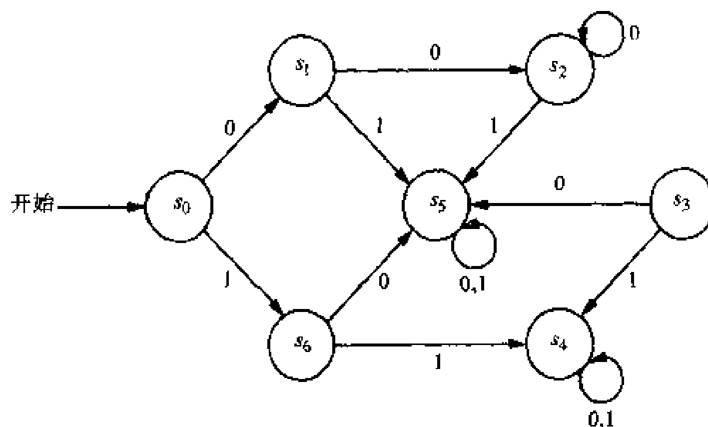
   a) $(\mathbf{0}^*\mathbf{1}^*)^*$　　b) $(\mathbf{0}(\mathbf{01}^*\mathbf{0})^*)^*$　　c) $(\mathbf{0}^*\cup(\mathbf{01})^*\cup\mathbf{1}^*)^*$

13. 构造一个带输出的有限状态机，若到目前为止读到的二进制数输入串中含有 4 个或 4 个以上的 1，它则输出 1。然后再构造一个确定型有限状态自动机来识别这个集合。

14. 构造一个带输出的有限状态机器，若到目前为止读到的二进制数输入串中含有 4 个或 4 个以上连续的 1，它则输出 1。然后再构造一个确定型有限状态自动机来识别这个集合。

15. 构造一个带输出的有限状态机器，若到目前为止读到的二进制数输入串以 4 个或 4 个以上连续的 1 结尾，它则输出 1。然后再构造一个确定型有限状态自动机来识别这个集合。

16. 在有限状态机中，称状态 s' 是从状态 s 可达的，如果存在输入串 x 使得 $f(s,x)=s'$。称状态 s 是瞬变的，若没有非空输入串 x 使得 $f(s,x)=s$。称状态 s 是一个沉积点，若对于任意输入串 x 都有 $f(s,x)=s$。对下列状态图所示的有限状态机，回答问题 a）到 d）。



a) 哪些状态是从 s_0 可达的？

b) 哪些状态是从 s_2 可达的？

c) 哪些状态是瞬变的？

d) 哪些状态是沉积点？

*17．设集合 S，I 和 O 都是有限集合，且 $|S|=n, |I|=k, |O|=m$。

a) 可以构造多少个不同的有限状态机（米利机）(S, I, O, f, g, s_0)（其中的初始状态 s_0 可以任意选择）？

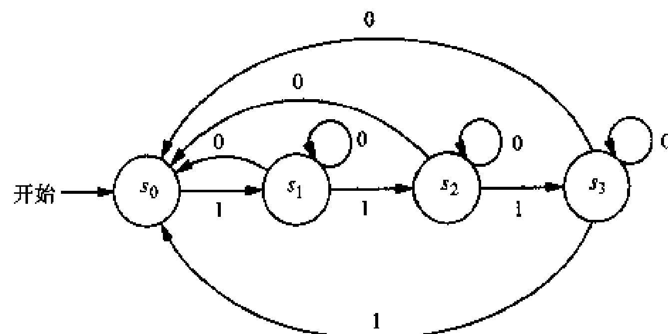
b) 可以构造多少个不同的摩尔机 (S, I, O, f, g, s_0)（其中的初始状态 s_0 可以任意选择）？

*18．设集合 S 和 I 是有限集合，且 $|S|=n, |I|=k$。在下列情形下，存在多少个不同的有限状态自动机（S，I，f，s_0，F）（其中的初始状态 s_0 以及由 S 的终结状态构成的子集 F 可以任意选择）？

a) 如果机器是确定型的。

b) 如果机器是非确定型的。（注：这包括确定型机器。）

19．对于具有如下状态图的非确定型自动机，构造一个与之等价的确定型有限状态自动机。



20．练习 19 中的自动机识别的语言是什么？

21．构造有限状态自动机识别下列集合：

a) $\mathbf{0}^*(\mathbf{10})^*$　　b) $(\mathbf{01}\cup\mathbf{111})^*\mathbf{10}^*(\mathbf{0}\cup\mathbf{1})$　　c) $(\mathbf{001}\cup(\mathbf{11})^*)^*$

*22 求表示由 0 和 1 组成的下列串集合的正则表达式：

a) 偶数个 1 与奇数个 0 交替出现。

b) 包含至少 2 个连续的 0 或 3 个连续的 1。

c) 不包含 3 个连续的 0 或 2 个连续的 1。

*23．试证：若 A 是一个正则集合，则 A 也是。

*24．试证：若 A 和 B 都是正则集合，则 $A\cap B$ 也是。

*25．求识别由 0 和 1 组成的下列串集合的有限状态自动机：

a) 以不超过 3 个连续的 0 开始，且至少包含 2 个连续的 1。

b) 包含偶数多个符号，且不含 101。

c) 有 3 个由 2 个或 2 个以上的 1 组成的块，且有至少 2 个 0。

*26 用 10.4 节的练习 16 所给的泵引理证明：$\{0^{2^n} | n\in\mathbf{Z}\}$ 不是正则的。

*27．用 10.4 节的练习 16 所给的泵引理证明：$\{1^p | p$ 是素数$\}$ 不是正则的。

*28．对于上下文无关语言，有与正则集合的泵引理类似的结果。设 $L(G)$ 是上下文无关语言 G 识别的语言。此结果是：存在常量 N，如果 z 是 $L(G)$ 中的一个词，且 $l(z)\geqslant N$，

则 z 可以写成 $uvwxy$，其中 $l(vwx) \leqslant N$，$l(vx) \geqslant 1$，且 uv^iwx^iy 属于 $L(G)(i=0,1,2,3,\cdots)$。用这个结果证明：不存在上下文无关文法 G 满足 $\{0^n1^n2^n \mid n=0,1,2,\cdots\}$。

计算机题目

写出具有下列输入和输出的程序。

1. 给定短语结构文法的产生式，根据乔姆斯基分类方法，判断此文法所在的类。
*2. 给定一个上下文无关文法的产生式和一个串，如果这个串在此文法生成的语言中，产生这个串的派生树。
3. 给定一个摩尔机的状态表和一个输入串，产生此机器生成的输出串。
4. 给定一个米利机的状态表和一个输入串，产生此机器生成的输出串。
5. 给定一个确定型有限状态自动机的状态表和一个串，判断这个串能否由此自动机识别。
6. 给定一个非确定型有限状态自动机的状态表和一个串，判断这个串能否由此自动机识别。
*7. 给定一个非确定型有限状态自动机的状态表，构造一个识别相同语言的确定型有限状态自动机的状态表。
**8. 给定一个正则表达式，构造一个非确定型有限状态自动机识别这个表达式表示的集合。
9. 给定一个正则文法，构造一个有限状态自动机识别这个文法生成的语言。
10. 给定一个有限状态自动机，构造一个正则文法生成这个自动机识别的语言。
*11. 给定一个图灵机，求一个给定的输入串所产生的输出串。

计算和研究

用一个计算程序或你已经写出的程序做下列练习。

1. 如下解两个状态的忙碌海狸问题：检查所有具有两个状态且字母表为 $\{1,B\}$ 的图灵机。
*2. 如下解 3 个状态的忙碌海狸问题：检查所有具有 3 个状态且字母表为 $\{1,B\}$ 的图灵机。
**3. 如下求 4 个状态的忙碌海狸机器：检查所有具有两个状态且字母表为 $\{1,B\}$ 的图灵机。
**4. 尽力解 5 个状态的忙碌海狸问题，进展越多越好。
**5. 尽力解 6 个状态的忙碌海狸问题，进展越多越好。

写作题目

用课本以外的资料解决下列问题，并写成短文。

1. 描述怎么用一个利登梅耶系统（Lidenmeyer system）来建立某种类型植物的生长模型。利登梅耶系统用带产生式的文法来建立植物生长的各种不同方式的模型。
2. 对于各种各样的程序设计语言，如 Java，LISP，ADA 和数据库语言 SQL，给出描述其语法的巴科斯-诺尔范式规则。
3. 解释在网络协议研究中，怎么使用有限状态机。
4. 解释概念“有限状态自动机极小化”。给出一个程序来实现这个极小化。
5. 给出细胞自动机的定义，解释它们的应用（以“生命的游戏”（Game of Life）为例）。
6. 定义下推自动机，解释怎么用下推自动机来识别集合。下推自动机能识别哪些的集合？给出验证你的答案正确性的证明概要。

7. 定义线性有界自动机，解释怎么用线性有界自动机来识别集合。线性有界自动机能识别哪些的集合？给出验证你的答案正确性的证明概要。
8. 查找图灵对现称为图灵机的机器的原始定义。他定义这样机器的动机是什么？
9. 描述“通用图灵机”（Universal Turing Machine）的概念。解释怎么构造这样的机器。
10. 解释能够用非确定型图灵机而不能用确定型图灵机的应用种类。
11. 证明一个图灵机能够模拟一个非确定型图灵机的任何动作。
12. 证明一个集合能被图灵机识别当且仅当它能由短语结构文法生成。
13. 描述 λ 演算的基本概念。解释怎么用它来研究函数的可计算性。
14. 试证：一个具有 n 个带的图灵机能做的任何事情，本章所定义的图灵机也都能做。
15. 试证：一个在两个方向都有无限带的图灵机能做的任何事情，只在一个方向有无限带的图灵机也都能做。

附录 A　指数函数和对数函数

本附录将复习指数函数和对数函数的一些基本性质。本教材从头到尾都要用到这些性质。需要对这些材料做更深入复习的学生，可以参考微积分教材或其先修教材，例如在“推荐读物”中提出的那些教材。

A.1　指数函数

设 n 是一个正整数，b 是一个固定的正实数，函数 $f_b(n)=b^n$ 是由

$$f_b(n)=b^n=b\cdot b\cdot b\cdot\cdots\cdot b$$

定义的，其中等式的右边是 n 个因子 b 相乘。

由微积分理论知：可以对任意实数 x 定义函数 $f_b(x)=b^x$。函数 $f_b(x)=b^x$ 称为以 b 为底的指数函数。对于以 b 为底的指数函数，当 x 不是整数时，其函数值的求法将略去。

下面的定理 1 给出了指数函数的两个重要性质。它们及其相关性质的证明可以在微积分教科书中找到。

定理 1　设 b 是一个实数，则

i) $b^{x+y}=b^xb^y$。

ii) $(b^x)^y=b^{xy}$。

图 1 是一些指数函数的函数图。

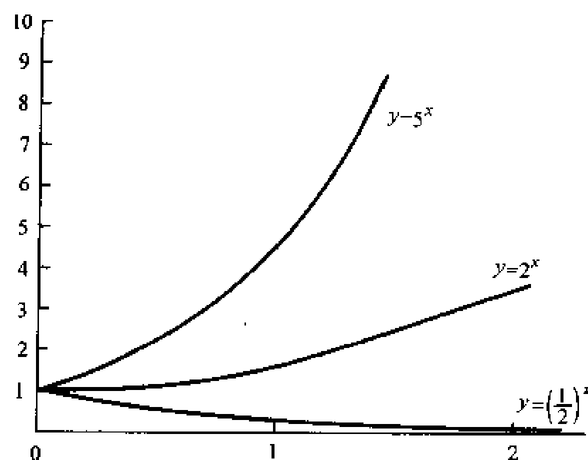


图 1　底为 $\frac{1}{2}$、2 和 5 的指数函数的图形

A.2　对数函数

设 b 是一个实数，且 $b>1$，则指数函数 b^x 是严格递增的（微积分证明的一个事实），且是从实数集到非负实数集的一个一一对应，因而就有反函数，此反函数称为以 b 为底的

对数函数。换句话说，如果 b 是一个大于 1 的实数，x 是一个正实数，则

$$b^{\log_b x} = x$$

这个函数在 x 处的值称为以 b 为底 x 的对数。

由定义可以看出

$$\log_b b^x = x$$

定理 2 给出了对数的几个重要性质。

定理 2　设 b 是一个大于 1 的实数，则

i) 当 x 和 y 都是正实数时，$\log_b(xy) = \log_b x + \log_b y$。

ii) 当 x 是正实数时，$\log_b(x^y) = y\log_b x$。

证　因为$\log_b(xy)$是满足 $b^{\log_b(xy)} = xy$ 的唯一实数，所以，为证明第 1 部分，只要证明 $b^{\log_b x + \log_b y} = xy$。根据定理 1 的第 1 部分，我们有

$$\begin{aligned} b^{\log_b x + \log_b y} &= b^{\log_b x} b^{\log_b y} \\ &= xy \end{aligned}$$

为证明第 2 部分，只要证明 $b^{y\log_b x} = x^y$。根据定理 1 的第 2 部分，我们有

$$\begin{aligned} b^{y\log_b x} &= (b^{\log_b x})^y \\ &= x^y \end{aligned}$$

□

下面定理将两个不同底的对数联系在一起。

定理 3　设 a 和 b 是大于 1 的实数，x 是正实数，则

$$\log_a x = \log_b x / \log_b a$$

证　为证明这个结果，只要证明

$$b^{\log_a x \cdot \log_b a} = x$$

根据定理 1 的第 2 部分，我们有

$$\begin{aligned} b^{\log_a x \log_b a} &= (b^{\log_b a})^{\log_a x} \\ &= a^{\log_a x} \\ &= x \end{aligned}$$

这就完成了证明。　□

本教材中，最常用的对数的底是 2，所以我们用记号 $\log x$ 来记 $\log_2 x$。

图 2 是函数 $f(x) = \log x$ 的图形。根据定理 3，如果底 b 不是 2，所得的函数是 $\log x$ 的常数倍，即$(1/\log b)\log x$。

练习

1. 将下列每个数表示成 2 的乘幂。

　a) $2 \cdot 2^2$　　b) $(2^2)^3$　　c) $2^{(2^2)}$

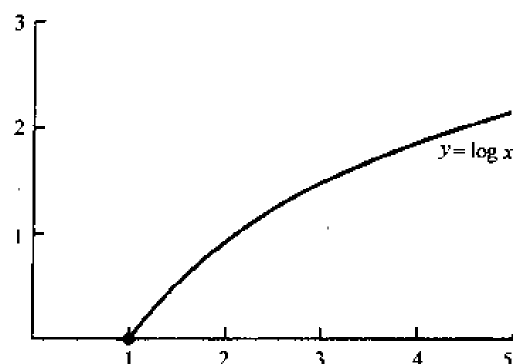


图 2 对数函数 $f(x)=\log x$ 的图形

2. 求下列各式的值。

a) $\log_2 1024$ b) $\log_2 1/4$ c) $\log_4 8$

3. 设 $\log_4 x = y$，求下列各式的值。

a) $\log_2 x$ b) $\log_8 x$ c) $\log_{16} x$

☞4. 设 a，b 和 c 是正实数。证明 $a^{\log_b c} = c^{\log_b a}$。

5. 当 b 为下列值时，画出 $f(x) = b^x$ 的图形。

a) 3 b) 1/3 c) 1

6. 当 b 为下列值时，画出 $f(x) = \log_b x$ 的图形。

a) 4 b) 100 c) 1000

附录B 伪 代 码

在本教材中，用英语和伪代码两种语言来描述算法。伪代码是过程步骤的英语描述和实际程序设计语言的过程说明之间的一个中间步骤。使用伪代码的优点在于它的简单性和可理解性，它很容易被写出来，也容易从它产生实际的计算机代码（用各种各样的程序设计语言）。

在这个附录中，我们描述本教材中使用的伪代码的格式和语法，这种伪代码的基本结构与 Pascal 的非常类似，而 Pascal 是目前最广泛用于教学中的一种程序设计语言。但我们使用的伪代码比正式的程序设计语言要自由得多，因为许多步骤都可以使用英语来描述。

本附录不是一个正式的学习材料，相反，应该把它当作学生的参考指南。当学生在学习课文中算法的描述时，或在写练习的伪代码求解时，可以使用它。

B.1 过程语句

算法的伪代码描述以语句 **procedure** 开始，这个语句给出算法的名称、列出输入变元和描述每个输入变元所具有的类型。例如，语句

procedure *maximum*(*L*:整数列表)

是一个算法的伪代码描述中的第一个语句，此算法的名称是 *maximum*，它要求一列整数的最大值。

B.2 赋值语句和其他语句类型

赋值语句是用来对变元进行赋值的。在赋值语句中，左边是变元名，右边是一个表达式，表达式中可以包含常量、已被赋值的变元和过程定义的函数。右边可以包含任何常见的算术运算。本书中的伪代码还可以包含任何已经定义的运算，即使这些运算要需用实际程序设计语言中的许多语句才能实现。

赋值用符号:=表示。这样，赋值语句的形式为

变元:=表达式

例如，语句

$max := a$

将 a 的值赋给变元 max。也可以使用如下语句：

x :=列表 L 中的最大整数

它将 x 的值置为列表 L 中的最大整数。要将这个语句翻译成实际程序设计语言，可能需要不止一个语句。同时，可以用命令

```
交换 a 和 b
```

来交换 a 和 b。虽然能够将这个语句表示成多个赋值语句（见本附录的练习 2），但为简单起见，我们经常使用这个伪代码的简写形式。

B.3 命令块

可以将许多语句分成块来执行复杂过程。这些块用 **begin** 和 **end** 语句来划分，块中的所有语句都按相同的间隔缩进。

```
begin
    语句 1
    语句 2
    语句 3
      ⋮
    语句 n
end
```

块中的所有语句按顺序执行。

B.4 注释

在本书的伪代码中，以花括号括起来的语句是不执行的，这样的语句被用作注释或提示，帮助解释过程是怎么运行的。例如，可以用语句

```
{x 是 L 中的最大元素}
```

来提醒读者，在过程的那个地方，x 等于列表 L 中的最大元素。

B.5 条件结构

我们将使用的最简单的条件结构是

```
if 条件 then 语句
```

或者

```
if 条件 then
begin
    语句块
end
```

执行时，首先检查条件，如果它为真，则执行所给的语句。例如，在 2.1 节的算法 1（它求一个整数集合的最大值）中，用条件语句来检查 $max < a_i$ 是否对每个变元都成立，如果是，就将 a_i 的值赋给 max。

常常要用更一般的（条件）结构。这种结构不仅在一种情形下（当所给的条件为真时）能够使用，在另一种情形下（条件为假）时也能使用。这种结构是

```
if 条件 then 语句 1
else 语句 2
```

注意，语句 1 和 2 都可以用一个程序块来替代。有时候，还要用更一般形式的条件结构，这种结构是

```
if 条件 1 then 语句 1
else if 条件 2 then 语句 2
else if 条件 3 then 语句 3
    ⋮
else if 条件 n then 语句 n
else 语句 n+1
```

在使用这种结构时，若条件 1 为真，则执行语句 1，然后程序就退出此结构。如果条件 1 为假，则程序检查条件 2 是否为真；如是，则执行语句 2；等等。这样，如果前面 $n-1$ 个条件都不成立，但条件 n 成立，则执行语句 n。最后如果条件 1、条件 2、条件 3、…、条件 n 都不成立，则执行语句 $n+1$。注意，$n+1$ 个语句中的任何一个都可用一个程序块来替代。

B.6 循环结构

本书的伪代码中，有两种循环结构，第一是“for 结构”，它的形式是

```
for 变元:=起始值  to  结束值
    语句
```

或

```
for 变元:=起始值  to  结束值
begin
    语句块
end
```

循环开始时，若起始值小于或等于结束值，则将起始值赋给变元，并在变元的这个值下，执行结构尾部的语句。然后变元的值增加 1，并在变元的这个新值下，再执行这个语句（或语句块中的语句)。这样一直重复下去，直至变元达到结束值。在执行这个命令之后，变元等于结束值，并且算法继续进行到下一个语句。如果起始值超过结束值，循环中的任何语句都不执行。

下列伪代码用“for”循环结构求正整数 1 到 n 的和。

```
sum := 0
for i := 1 to n
    sum := sum + 1
```

本书中还用一种更一般的“for”语句，其形式是

```
for 具有某种性质的所有元素
```

它的含义是：对具有某种性质的所有元素，跟在它后面的语句或语句组将相继被执行。

我们使用的第二类循环结构是“while”结构，其形式是

```
while 条件
    语句
```

或

```
while 条件
begin
    语句块
end
```

在执行这个结构时，首先检查所给的条件，若为真，则执行其后面的语句，这次执行可能改变某些变元的值，而这些变元是条件的一部分。在这些命令执行完毕后，若条件仍然为真，则再执行这些语句。此过程一直执行到条件变成假为止。例如，可以用下列包含“while的伪代码段来求整数 1 到 n 的和。

```
sum := 0
while n >0
begin
    sum := sum + n
    n := n - 1
end
```

注意，任何“for”结构都可转换为“while”结构（见本附录结尾的练习 3），但“for”结构更容易理解。所以，只要适合时，就优先使用“for”结构而不是对应的“while”结构。

B.7 在过程中使用其他过程

在一个过程内部还可以使用其他过程（或在递归程序中使用其自身），方法是写出这个过程的名称，后面再跟上此过程的输入。例如：

```
max(L)
```

将执行输入为 L 的过程 max。当这个过程中的所有步骤都执行完毕后，再继续执行这个过程的下一个语句。

练习

1. 下列两个赋值语句块的区别是什么？

   $a := b$

   $b := c$

和

$b := c$

$a := b$

2. 用赋值语句构造一个过程来交换变元 x 和 y 的值。所需赋值语句的最少个数是什么？
3. 说明怎样用 "while" 结构来写形式为

for i :=起始值 **to** 结束值

语句

的循环？

奇数练习题答案

第 1 章

1.1 节

1. a) 是，T　b) 是，F　c) 是，T　d) 是，F
   e) 不是　f) 不是　g) 是，T
3. a) 今天不是星期四。
   b) 新泽西州有污染。
   c) $2+1\neq 3$。
   d) 缅因州的夏天不热或阳光不明媚。
5. a) $p\wedge q$　b) $p\wedge\neg q$　c) $\neg p\wedge\neg q$　d) $p\vee q$
   e) $p\rightarrow q$　f) $(p\vee q)\wedge(p\rightarrow\neg q)$　g) $q\leftrightarrow p$
7. a) $\neg p$　b) $p\wedge\neg q$　c) $p\rightarrow q$　d) $\neg p\rightarrow\neg q$
   e) $p\rightarrow q$　f) $q\wedge\neg p$　g) $q\rightarrow p$
9. a) False（假）　b) True（真）　c) True（真）　d) True（真）
   e) True（真）　f) True（真）　g) False（假）　h) True（真）
11. a) 同或：如果你学过微积分或计算机科学，或两者都学过，就可以选修离散数学。异或：如果你学过微积分或计算机科学，但并非两者都学过，就可以选修离散数学。这里很可能想表示的是同或。
    b) 同或：你可以拿回扣，也可以得低息贷款，你也可以既拿回扣又得低息贷款。异或：你可以拿回扣，也可以得低息贷款，但不能既拿回扣，又得低息贷款。这里很可能想表示的是异或。
    c) 同或：你可以从 A 列选两项，不选 B 列；你也可以不选 A 列而从 B 列选三项；你还可以共选五项，包括 A 列两项和 B 列三项。异或：你可以从 A 列选两项，也可以从 B 列选三项，但不能都选。这里很可能想表示的是异或。
    d) 同或：2 英尺多的雪或冷风零下 100 度（华氏），或两者均具备，学校将停课。异或：2 英尺多的雪或冷风零下 100 度，但并非两者均具备，学校将停课。这里很可能想表示的是同或。
13. “如果我问你右边的路是否通向废墟，你会说是吗?”
15. a) 东北风吹就下雪。
    b) 温暖若能持续一周，苹果树就会开花。
    c) 若活塞队赢得冠军，那么他们打败了湖人队。
    d) 如果你登上了朗峰顶，那你必定已走了 8 英里（1 英里 = 1.6 公里）。
    e) 如果你世界闻名，就能做终身教授。

f）如果你驾车超过 400 英里，就得买汽油了。

g）如果你的保修单有效，你购买 CD 机必定还不足 90 天。

17. a）当且仅当外边很热时你才能买冰激淋卷。

b）当且仅当你手持唯一的胜券时才能赢得比赛。

c）当且仅当你有关系时才会得到提拔。

d）当且仅当你看电视时思想才会衰退。

e）当且仅当我乘坐火车的日子，它才会晚点。

19. a）逆命题："只有今天下雪，我明天才去滑雪。"换位命题："如果我明天不滑雪，那么今天一定没有下雪。"

b）逆命题："如果我今天来上课，就会有测验。"换位命题："如果我今天不来上课，就不会有测验。"

c）逆命题："如果一个正整数没有 1 和它自身以外的除数，它就是素数。"换位命题："如果一个正整数有既不是 1 又不是它自己的除数，它不是素数。"

21.

a)

p	$\neg p$	$p \wedge \neg p$
T	F	F
F	T	F

b)

p	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

c)

p	q	$\neg q$	$p \vee \neg q$	$(p \vee \neg q) \to q$
T	T	F	T	T
T	F	T	T	F
F	T	F	F	T
F	F	T	T	F

d)

p	q	$p \vee q$	$p \wedge q$	$(p \vee q) \to (p \wedge q)$
T	T	T	T	T
T	F	T	F	F
F	T	T	F	F
F	F	F	F	T

e)

p	q	$p \to q$	$\neg q$	$\neg p$	$\neg q \to \neg p$	$(p \to q) \leftrightarrow (\neg q \to \neg p)$
T	T	T	F	F	T	T
T	F	F	T	F	F	T
F	T	T	F	T	T	T
F	F	T	T	T	T	T

f)

p	q	$p \to q$	$q \to p$	$(p \to q) \to (q \to p)$
T	T	T	T	T
T	F	F	T	T
F	T	T	F	F
F	F	T	T	T

23.

a)

p	q	$p \leftrightarrow \neg q$
T	T	F
T	F	T
F	T	T
F	F	T

b)

p	q	$\neg p \leftrightarrow q$
T	T	F
T	F	T
F	T	T
F	F	F

c)

p	q	$(p \to q) \vee (\neg p \to q)$
T	T	T
T	F	T
F	T	T
F	F	T

d)

p	q	$(p \to q) \wedge (\neg p \to q)$
T	T	T
T	F	F
F	T	T
F	F	F

e)

p	q	$(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$
T	T	T
T	F	T
F	T	T
F	F	T

f)

p	q	$(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$
T	T	T
T	F	T
F	T	T
F	F	T

25.

a)

p	q	r	$p \to (\neg q \vee r)$
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

b)

p	q	r	$\neg p \to (q \to r)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	T

c)

p	q	r	$(p\to q)\vee(\neg p\to r)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

d)

p	q	r	$(p\to q)\wedge(\neg p\to r)$
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

e)

p	q	r	$(p\leftrightarrow q)\vee(\neg q\leftrightarrow r)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	T
F	F	F	T

f)

p	q	r	$(\neg p\leftrightarrow\neg q\leftrightarrow(q\leftrightarrow r)$
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

27.

p	q	r	s	$p\leftrightarrow q$	$r\leftrightarrow s$	$(p\leftrightarrow q)\leftrightarrow(r\leftrightarrow s)$
T	T	T	T	T	T	T
T	T	T	F	T	F	F
T	T	F	T	T	F	F
T	T	F	F	T	T	T
T	F	T	T	F	T	F
T	F	T	F	F	F	T
T	F	F	T	F	F	T
T	F	F	F	F	T	F
F	T	T	T	F	T	F
F	T	T	F	F	F	T
F	T	F	T	F	F	T
F	T	F	F	F	T	F
F	F	T	T	T	T	T
F	F	T	F	T	F	F
F	F	F	T	T	F	F
F	F	F	F	T	T	T

29. a) 按位或是 111 1111；按位并是 000 0000；按位异或（按位加）是 111 1111。
    b) 按位或是 1111 1010；按位并是 1010 0000；按位异或（按位加）是 0101 1010。
    c) 按位或是 10 0111 1001；按位并是 00 0100 0000；按位异或（按位加）是 10 0011 1001。
    d) 按位或是 11 1111 1111 ；按位并是 00 0000 0000；按位异或（按位加）是 11 1111 1111。
31. 0.2，0.6
33. 0.8，0.6
35. 不一致。
37. NEW **AND** JERSEY **AND** BEACHES，(JERSEY **AND** BEACHES) **NOT** NEW
39. 按薪水减少的顺序：傅雷德、麦吉、杰尼斯。
41. 侦探可以断定男管家和厨师说谎，但不能判断究竟是园丁还是杂役说真话。

1.2 节

1. 等价关系由下表中相应两列的一致推出。

p	$p\wedge T$	$p\vee F$	$p\wedge F$	$p\vee T$	$p\vee p$	$p\wedge p$
T	T	T	F	T	T	T
F	F	F	F	T	F	F

3.
a)

p	q	$p\vee q$	$q\vee p$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

b)

p	q	$p\wedge q$	$q\wedge p$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

5.

p	q	r	$q\vee r$	$p\wedge(q\vee r)$	$p\wedge q$	$p\wedge r$	$(p\wedge q)\vee(p\wedge r)$
T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T
T	F	T	T	T	F	T	T
T	F	F	F	F	F	F	F
F	T	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	F	T	T	F	F	F	F
F	F	F	F	F	F	F	F

7.

a)

p	q	$p\wedge q$	$(p\wedge q)\to p$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

b)

p	q	$p\vee q$	$p\to(p\vee q)$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	T

c)

p	q	$\neg p$	$p\to q$	$\neg p\to(p\to q)$
T	T	F	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

d)

p	q	$p\wedge q$	$p\to q$	$(p\wedge q)\to(p\to q)$
T	T	T	T	T
T	F	F	F	T
F	T	F	T	T
F	F	F	T	T

e)

p	q	$p\to q$	$\neg(p\to q)$	$\neg(p\to q)\to p$
T	T	T	F	T
T	F	F	T	T
F	T	T	F	T
F	F	T	F	T

f)

p	q	$p\to q$	$\neg(p\to q)$	$\neg q$	$\neg(p\to q)\to\neg q$
T	T	T	F	F	T
T	F	F	T	T	T
F	T	T	F	F	T
F	F	T	F	T	T

9. 对每种情况都证明只要前提成立，结论就成立。

a）如果前题 $p\wedge q$ 成真，那么根据合取的定义，结论 p 必为真。

b）如果前题 p 成真，根据析取的定义，结论 $p\vee q$ 也为真。

c）如果前题 $\neg p$ 成真，亦即 p 为假，那么结论 $p\to q$ 为真。

d）如果前题 $p\wedge q$ 为真，那么 p 和 q 均为真，所以结论 $p\to q$ 为真。

e）如果前题 $\neg(p\to q)$ 为真，那么 $p\to q$ 为假，所以 p 为真（同时 q 为假）。

f）如果前题 $\neg(p\to q)$ 为真，那么 $p\to q$ 为假，所以 p 为真而 q 为假。因而结论 $\neg q$ 为真。

11. a）如果 p 为真，那么 $p\vee(p\wedge q)$ 为真，因为析取中第一项为真。另一方面，如果 p 为假，那么 $p\wedge q$ 为假，所以 $p\vee(p\wedge q)$ 也为假。由于 p 和 $p\vee(p\wedge q)$ 总是有同样的真值，它们等价。

b）如果 p 为假，那么 $p\wedge(p\vee q)$ 也为假，因为合取中第一项为假。另一方面，如果 p 为真，那么由于 $p\vee q$ 为真，所以合取的两项均为真。由于 p 和 $p\wedge(p\vee q)$ 总是有同样的真值，它们等价。

13. 唯一能使这一蕴含关系为假的情况是 $\neg q\wedge(p\to q)$ 为真而 $\neg p$ 为假。由 $\neg p$ 为假知 p 必为真。由 $\neg q\wedge(p\to q)$ 为真，知 $\neg q$ 必为真，从而 q 为假。由 p 为真知 $p\to q$ 必为假，而这是不可能的。

15. 这些命题不可能逻辑等价，因为当 p、q 和 r 全为假时，$(p\to q)\to r$ 也为假，但 $p\to(q\to r)$ 为真。

17. 命题 $\neg p\leftrightarrow q$ 为真当且仅当 $\neg p$ 和 q 有同样的真值，也就是 p 和 q 有不同的真值。类似地，$p\leftrightarrow\neg q$ 在完全同样的情况下为真。所以这两个表达式逻辑等价。

19. 命题¬(p↔q)为真当且仅当p↔q为假，即p和q有不同的真值。由于这恰恰就是¬p↔q为真的情况，所以这两个表达式逻辑等价。
21. 如果取两次对偶，那么每个∨先变成∧，又变回∨；每个∧先变成∨，又变回∧；每个T先变为F，再变回T；每个F先变为T，再变回F。因此$(s^*)^*=s$。
23. 令p和q为只含运算符∧、∨和¬，以及T和F的等价复合命题，注意¬p和¬q也等价。反复用德摩根定律尽可能把¬往这些复合命题中推进，同时改∨为∧，改∧为∨，改T为F，改F为T。这样能证明¬p和¬q与p^*和q^*是一样的，只是其中的原子命题被它的非命题所代替。由此可断定p^*和q^*也等价，因为¬p和¬q等价。
25. $(p\wedge q\wedge\neg r)\vee(p\wedge\neg q\wedge r)\vee(\neg p\wedge q\wedge r)$
27. 对给定的复合命题p，构造其真值表，然后用析取范式写下一个逻辑等价于p的命题q。由于q只涉及¬、∧和∨，这就证明了这三个运算符构成一个功能完全集。
29. 根据练习27，给定一个复合命题p，我们可以写出一个只含¬、∧和∨，且与p逻辑等价的命题q。根据德摩根定律，我们可以用$\neg(\neg p_1\vee\neg p_2\vee\cdots\vee\neg p_n)$取代$(p_1\wedge p_2\wedge\cdots\wedge p_n)$，从而消去所有∧。
31. 在p或q或两者均为假时，$\neg(p\wedge q)$为真；当p和q均为真时，$\neg(p\wedge q)$为假。由于这就是$p\mid q$的定义，所以这两个复合命题逻辑等价。
33. $\neg(p\vee q)$在p和q均为假时为真，否则为假。由于这是$p\downarrow q$的定义，两者逻辑等价。
35. $((p\downarrow p)\downarrow q)\downarrow((p\downarrow p)\downarrow q)$
37. 从真值表或$p\mid q$的定义可直接得出结论。
39. 16
41. 若数据库是打开的，那么或者系统置于初始状态，或者监督程序置于关闭状态。

1.3节

1. a) T　b) T　c) F
3. a) T　b) F　c) F　d) F
5. a) 有个学生除周末外每天都花5个多小时在课堂上。
   b) 所有学生除周末外每天都花5个多小时在课堂上。
   c) 有个学生并非除周末外每天都花5个多小时在课堂上。
   d) 没有学生除周末外每天都花5个多小时在课堂上。
7. a) Sarah Smith已经访问过www.att.com。
   b) 至少一个人已访问过www.imdb.org。
   c) Joes Orez已经至少访问了一个网站。
   d) 至少有一个网站是Ashok Puri和Cindy Yoon都访问过的。
   e) 除David Belcher以外还有一个人访问过David Belcher已访问过的所有网站。
   f) 有两个不同的人已访问过完全同样的网站。
9. a) $\exists x\,(P(x)\wedge Q(x))$
   b) $\exists x\,(P(x)\wedge\neg Q(x))$
   c) $\forall x\,(P(x)\vee Q(x))$
   d) $\forall x\,\neg(P(x)\vee Q(x))$

11. a) $\forall x L(x, \text{Jerry})$
   b) $\forall x \exists y L(x, y)$
   c) $\exists y \forall x L(x, y)$
   d) $\forall x \exists y \neg L(x, y)$
   e) $\exists x \neg L(\text{Lydia}, x)$
   f) $\exists x \forall y \neg L(y, x)$
   g) $\exists x(\forall y L(y, x) \wedge \forall z((\forall w L(w, z)) \rightarrow z = x))$
   h) $\exists x \exists y(x \neq y \wedge L(\text{Lynn}, x) \wedge L(\text{Lynn}, y) \wedge \forall z(L(\text{Lynn}, z) \rightarrow (z = x \vee z = y)))$
   i) $\forall x L(x, x)$
   j) $\exists x \forall y(L(x, y) \leftrightarrow x = y)$
13. a) $A(\text{Lois}, \text{Michaels})$
   b) $\forall x(S(x) \rightarrow A(x, \text{Gross}))$
   c) $\forall x(F(x) \rightarrow (A(x, \text{Miller}) \vee A(\text{Miller}, x)))$
   d) $\exists x(S(x) \wedge \forall y(F(y) \rightarrow \neg A(x, y)))$
   e) $\exists x(F(x) \wedge \forall y(S(y) \rightarrow \neg A(y, x)))$
   f) $\forall y(F(y) \rightarrow \exists x(S(x) \vee A(x, y)))$
   g) $\exists x(F(x) \wedge \forall y((F(y) \wedge (y \neq x)) \rightarrow A(x, y)))$
   h) $\exists x(S(x) \wedge \forall y(F(y) \rightarrow \neg A(y, x)))$
15. a) $\neg M(\text{Chou}, \text{Koko})$
   b) $\neg M(\text{Arlene}, \text{Sarah}) \wedge \neg T(\text{Arlene}, \text{Sarah})$
   c) $\neg M(\text{Deborah}, \text{Jose})$
   d) $\forall x M(x, \text{ken})$
   e) $\forall x \neg T(x, \text{Nina})$
   f) $\forall x(T(x, \text{Avi}) \vee M(x, \text{Avi}))$
   g) $\exists x \forall y(y \neq x \rightarrow M(x, y))$
   h) $\exists x \forall y(y \neq x \rightarrow (M(x, y) \vee T(x, y)))$
   i) $\exists x \exists y(x \neq y \wedge M(x, y) \wedge M(y, x))$
   j) $\exists x M(x, x)$
   k) $\exists x \forall y(x \neq y \rightarrow (\neg M(x, y) \wedge \neg T(y, x)))$
   l) $\forall x(\exists y(x \neq y \wedge (M(y, x) \vee T(y, x))))$
   m) $\exists x \exists y(x \neq y \wedge M(x, y) \wedge T(y, x))$
   n) $\exists x \exists y(x \neq y \wedge \forall z((z \neq x \wedge z \neq y) \rightarrow (M(x, z) \vee M(y, z) \vee T(x, z) \vee T(y, z))))$
17. a) $\forall x P(x)$，其中$P(x)$为“x 需要一门离散数学课”，论域是所有计算机科学的学生集合。
   b) $\exists x P(x)$，其中 $P(x)$为“x 有台PC（计算机）”，论域是班上所有学生的集合。
   c) $\forall x \exists y P(x, y)$，其中$P(x, y)$为“$x$ 选修 y”，x 的论域是班上所有学生的集合，y 的论域是计算机科学课程的集合。
   d) $\exists x \exists y P(x, y)$，其中 $P(x, y)$ 以及 x、y 的论域同 c)。

e) $\forall x \forall y P(x,y)$，其中 $P(x,y)$ 为"x 去过 y"，x 的论域是班上所有学生的集合，y 的论域是校园内所有楼房的集合。

f) $\exists x \exists y \forall z (P(z,y) \rightarrow Q(x,z))$，其中 $P(z,y)$ 为"z 在 y 中"，$Q(x,z)$ 为"x 去过 z"，其中 x 的论域是班上所有学生的集合，y 的论域是校园内所有楼房的集合，z 的论域则是房间的集合。

g) $\forall x \forall y \exists z (P(z,y) \land Q(x,z))$，其中 P、Q 及 x、y、z 的论域与 f) 相同。

19. a) T　b) T
c) F　d) F
e) T　f) F

21. a) True　b) False　c) True
d) True　e) True　f) True
g) True　h) True　i) False
j) False　k) True　l) False

23. a) $p(1,3) \lor p(2,3) \lor p(3,3)$
b) $p(1,1) \land p(1,2) \land p(1,3)$
c) $p(1,1) \land p(1,2) \land p(1,3) \land p(2,1) \land p(2,2) \land p(2,3) \land p(3,1) \land p(3,2) \land p(3,3)$
d) $p(1,1) \lor p(1,2) \lor p(1,3) \lor p(2,1) \lor p(2,2) \lor p(2,3) \lor p(3,1) \lor p(3,2) \lor p(3,3)$
e) $(p(1,1) \land p(1,2) \land p(1,3)) \lor (p(2,1) \land p(2,2) \land p(2,3)) \lor (p(3,1) \land p(3,2) \land p(3,3))$
f) $(p(1,1) \lor p(2,1) \lor p(3,1)) \land (p(1,2) \lor p(2,2) \lor p(3,2)) \land (p(1,3) \lor p(2,3) \lor p(3,3))$

25. a) $\exists x \exists y \neg P(x,y)$
b) $\exists y \forall x \neg P(x,y)$
c) $\exists y \exists x (\neg P(x,y) \land \neg Q(x,y))$
d) $(\forall x \forall y P(x,y)) \lor (\exists x \exists y \neg Q(x,y))$
e) $\exists x (\forall y \exists z \neg p(x,y,z) \lor \forall z \exists y \neg p(x,y,z))$

27. a) 有条会说话的狗。
b) 班上有人懂法语和俄语。
c) 班上有这样的同学，在任何两门不同的数学课中，要么这两门他都没选，要么他就选了这两门。
d) 每个人要么已访问过利比亚，要么从未访问过利比亚以外的任何国家。
e) 有人已攀登过西马拉雅山的每座山峰。
f) 有个人既没和 Kevin Bacon 出演同一部电影，也没和任何与 Kevin Bacon 出演过电影的人出演同一部电影。

29. $\forall x \forall y \forall z ((x \cdot y) \cdot z = x \cdot (y \cdot z))$

31. a) $\forall x (P(x) \rightarrow \neg Q(x))$
b) $\forall x (Q(x) \rightarrow R(x))$

c) $\forall x(P(x) \to \neg R(x))$

d) 得不出结论。也许有爱虚荣的教授，因为前提并不排除在无知者之外还有爱虚荣的人。

33. a) $\forall x(P(x) \to \neg Q(x))$

b) $\forall x(R(x) \to \neg S(x))$

c) $\forall x(\neg Q(x) \to S(x))$

d) $\forall x(P(x) \to \neg R(x))$

e) 结论成立。假定 x 是个婴儿，那么根据第 1 个前提，x 是缺乏逻辑的。于是由第 3 个前提，x 受歧视。第 2 个前提说如果 x 能管住鳄鱼就不受歧视，所以 x 管不住鳄鱼。

35. $\neg(\exists x \forall y p(x,y)) \leftrightarrow \forall x(\neg \forall y p(x,y)) \leftrightarrow \forall x \exists y \neg p(x,y)$

37. x 论域中至少有一个值使$P(x)$和 $Q(x)$中至少一个成真时，两个语句均为真。

39. a) 若 A 为真，两边都逻辑等价于$\forall x p(x)$。若 A 为假，左边显然为假。此外，对每个 x，$p(x) \wedge A$ 为假，所以右边为假。因此两边逻辑等价。

b) 若 A 为真，两边均逻辑等价于$\exists x P(x)$。若 A 为假，左边显然为假。此外，对每个 x，$P(x) \wedge A$ 为假。所以$\exists x(P(x) \wedge A)$为假。因此两边逻辑等价。

41. 为证明不是逻辑等价，令 $P(x)$为语句“x 为正”，$Q(x)$为语句“x 为负”，论域为整数集合。于是$\exists x P(x) \wedge \exists x Q(x)$为真，但$\exists x(P(x) \wedge Q(x))$为假。

43. a) 假定$\forall x P(x) \wedge \exists x Q(x)$为真，那么对所有 x，$p(x)$为真，而且有个元素 y 使$Q(y)$为真。因为$P(x) \wedge Q(y)$对所有 x 成真，而且有 y 使$Q(y)$成真，$\forall x \exists y (P(x) \wedge Q(y))$为真。反过来，假定第 2 个命题为真。令 x 为论域中的一个元素。有元素 y 使$Q(y)$为真，所以$\exists x Q(x)$为真。由于$\forall x P(x)$也为真，所以第 1 个命题也为真。

b) 假定$\forall x P(x) \vee \exists x Q(x)$为真。那么要么对所有 x，$P(x)$为真；要么有 y，使$Q(y)$为真。在第 1 种情况下，$P(x) \vee Q(y)$对所有 x 成真，所以$\forall x \exists y(P(x) \vee Q(y))$为真。在第 2 种情况下，对特定的 y，$Q(y)$成真，所以$P(x) \vee Q(y)$对所有 x 成真，从而$\forall x \exists y(P(x) \vee Q(y))$成真。反过来，假定第 2 个命题为真。如果 $P(x)$对所有 x 为真，那么第 1 个命题为真；否则 $P(x)$对某个 x 为假，对这一 x 必有 y 使$P(x) \vee Q(y)$成真。因此 $Q(y)$必成真，于是$\exists y Q(y)$为真。这样第 1 个命题为真。

45. a) True

b) False，除非论域只含一个元素。

c) True

47. $\exists x p(x) \wedge \forall x \forall y((p(x) \wedge p(y)) \to x = y)$

49. 我们将要说明，在表达式的子表达式均可转换为前束范式（PNF）的情况下，怎样把这个表达式转换为前束范式。这样从内向外就可以用这一方法把所有表达式转换为前束范式。（要使论证形式化，就得使用集合上的数学归纳法，这一方法在 3.3 节讨论。）由 1.2 节的练习 29 可知，可以假定命题中只使用逻辑联结符 $\vee$ 和 $\neg$。注意不带量词的命题已经是前束范式的形式。（这是论证的基本情况。）现在假定命题是 $QxP(x)$形的，其中

Q 为量词。因为 $P(x)$ 是比原命题短的表达式，它可以转换为 PNF。Qx 后面跟上这 PNF 仍是 PNF，它等价于原命题。其次，假定命题为 $\neg P$。如果 P 已经是 PNF 形式的，我们可以用表 3 的等价关系把否定符号 $\neg$ 移到所有量词内层。最后，假定命题为 $P \vee Q$ 形的，其中 P 和 Q 均为 PNF。如果只有 P 和 Q 中的一个含量词，那么用练习 38 可以把量词移到它们两个的前面。如果 P 和 Q 均含量词，可以用练习 37、42 或 43b）重写 $P \vee Q$，使两个量词都出现在形为 $R \vee S$ 的命题之前，然后再把 $R \vee S$ 转换为 PNF。

51. $\forall L \exists \varepsilon \forall \delta \exists x(0<|x-a|<\delta \wedge |f(x)-L| \geqslant \varepsilon)$

53. $\forall L \exists \varepsilon \forall N \exists n(n>N \wedge |a_n - L| \geqslant \varepsilon)$

55. $\forall \varepsilon(\forall N \exists n(n>N \wedge a_n > L-\varepsilon) \wedge \exists N \forall n(n>N \rightarrow a_n \leqslant L+\varepsilon))$

1.4 节

1. a) $\{-1,1\}$
   b) $\{1,2,3,4,5,6,7,8,9,10,11\}$
   c) $\{0,1,4,9,16,25,36,49,64,81\}$
   d) $\varnothing$

3. a) 相等　b) 不等　c) 不等

5. a) 是　b) 不是　c) 是　d) 不是　e) 不是　f) 不是

7. a) True　b) True　c) False
   d) True　e) True　f) False

9. 假定 $x \in A$。由于 $A \subseteq B$，则 $x \in B$。由于 $B \subseteq C$，则 $x \in C$。由 $x \in A$ 导出 $x \in C$，所以 $A \subseteq C$。

11. a) 1　b) 1　c) 2　d) 3

13. a) $\{\varnothing,\{a\}\}$　b) $\{\varnothing,\{a\},\{b\},\{a,b\}\}$
    c) $\{\varnothing,\{\varnothing\},\{\{\varnothing\}\},\{\varnothing,\{\varnothing\}\}\}$

15. a) 8　b) 16　c) 2

17. a) $\{(a,y),(b,y),(c,y),(d,y),(a,z),(b,z),(c,z),(d,z)\}$
    b) $\{(y,a),(y,b),(y,c),(y,d),(z,a),(z,b),(z,c),(z,d)\}$

19. 三元组 (a,b,c) 的集合，其中 a 是航线，b 和 c 是城市。

21. $\varnothing \times A = \{(x,y) | x \in \varnothing \wedge y \in A\} = \varnothing = \{(x,y) | x \in A \wedge y \in \varnothing\} = A \times \varnothing$

23. mn

25. 我们必须证明 $\{\{a\},\{a,b\}\} = \{\{c\},\{c,d\}\}$ 成立的充分必要条件是 $a=c$ 且 $b=d$。显然条件是充分的。为证必要性，假定这两个集合相等。首先考虑 $a \neq b$ 的情况，这时 $\{\{a\},\{a,b\}\}$ 恰含两个元素，其中的一个包含一个元素。这样 $\{\{c\},\{c,d\}\}$ 也必须有同样的性质，所以 $c \neq d$，且 $\{c\}$ 是两个元素中只含一个元素的那一个。于是 $\{a\}=\{c\}$，也就是 $a=c$。含两个元素的集合 $\{a,b\}$ 和 $\{c,d\}$ 也必须相等。由于 $a=c$，且 $a \neq b$，所以 $b=d$。其次假定 $a=b$。于是 $\{\{a\},\{a,b\}\} = \{\{a\}\}$，这是只含一个元素的集合。于是 $\{\{c\},\{c,d\}\}$ 也只含一个元素，这只有在 $c=d$ 时才有可能，此时该集合为 $\{\{c\}\}$。从而 $a=c$ 且 $b=d$。

27. 令 $S=\{a_1, a_2, \cdots, a_n\}$。把 S 的每个子集 S' 都用长度为 n 的位串表示，位串中第 i

位为 1 的充分必要条件是 $a_i \in S$。为产生 S 的所有子集，列出长度为 n 的所有 2^n 个位串（例如，按增序），再写下相应的子集。

1.5 节

1. a）住处离校不超过 1 英里且走路上学的学生集合。
   b）住处离校不超过 1 英里或走路上学的学生之集合。
   c）住处离校不超过 1 英里但不走路上学的学生集合。
   d）走路上学但住处离校超过 1 英里的学生集合。

3. a）$\{0, 1, 2, 3, 4, 5, 6\}$　　b）$\{3\}$
   c）$\{1, 2, 4, 5\}$　　d）$\{0, 6\}$

5. $\overline{\overline{A}} = \{x \mid \neg(x \in \overline{A})\} = \{x \mid \neg(\neg x \in A)\} = \{x \mid x \in A\} = A$。

7. a）$A \cup B = \{x \mid x \in A \vee x \in B\}$
   $= \{x \mid x \in B \vee x \in A\} = B \cup A$
   b）$A \cap B = \{x \mid x \in A \wedge x \in B\}$
   $= \{x \mid x \in B \wedge x \in A\} = B \cap A$

9. a）$x \in (\overline{A \cup B}) \Leftrightarrow x \notin (A \cup B) \Leftrightarrow \neg(x \in A \vee x \in B)$
   $\Leftrightarrow \neg(x \in A) \wedge \neg(x \in B) \Leftrightarrow x \notin A \wedge x \notin B \Leftrightarrow (x \in \overline{A}) \wedge (x \in \overline{B}) \Leftrightarrow x \in (\overline{A} \cap B\ \overline{B})$
   b）

A	B	$A \cup B$	$(\overline{A \cup B})$	$\overline{A}$	$\overline{B}$	$\overline{A} \cap \overline{B}$
1	1	1	0	0	0	0
1	0	1	0	0	1	0
0	1	1	0	1	0	0
0	0	0	1	1	1	1

11. a）$x \in \overline{A \cap B \cap C} \Leftrightarrow \notin A \cap B \cap C \Leftrightarrow x \notin A \vee x \notin B \cup x \notin C \Leftrightarrow x \in \overline{A} \vee x \in \overline{B} \vee x \in \overline{C} \Leftrightarrow x \in (\overline{A} \cup \overline{B} \cup \overline{C})$
   b）

$A\ B\ C$	$A \cap B \cap C$	$(\overline{A \cap B \cap C})$	$\overline{A}\ \overline{B}\ \overline{C}$	$\overline{A} \cup \overline{B} \cup \overline{C}$
1 1 1	1	0	0 0 0	0
1 1 0	0	1	0 0 1	1
1 0 1	0	1	0 1 0	1
1 0 0	0	1	0 1 1	1
0 1 1	0	1	1 0 0	1
0 1 0	0	1	1 0 1	1
0 0 1	0	1	1 1 0	1
0 0 0	0	1	1 1 1	1

13. 两边都等于 $\{x \mid x \in A \wedge x \notin B\}$。

15. a）$x \in A \cup (B \cup C) \Leftrightarrow (x \in A) \vee (x \in (B \cup C)) \Leftrightarrow (x \in A) \vee (x \in B \vee x \in C) \Leftrightarrow (x \in A \vee x \in B) \vee (x \in C) \Leftrightarrow x \in (A \cup B) \cup C$
   b）只需将 a）中 $\cup$ 替换成 $\cap$，$\vee$ 替换成 $\wedge$。

c) $x \in A \cup (B \cap C) \Leftrightarrow (x \in A) \vee (x \in (B \cap C)) \Leftrightarrow (x \in A) \vee (x \in B \wedge x \in C) \Leftrightarrow (x \in A \vee x \in B) \wedge (x \in A \vee x \in C) \Leftrightarrow x \in (A \cup B) \cap (A \cup C)$

17. a) {4, 6}
    b) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
    c) {4, 5, 6, 8, 10}
    d) {0, 2, 4, 5, 6, 7, 8, 9, 10}
19. a) $B \subseteq A$
    b) $A \subseteq B$
    c) $A \cap B = \emptyset$
    d) 没什么可说，因为它总是成真。
    e) $A = B$
21. $A \subseteq B \Leftrightarrow \forall x(x \in A \to x \in B) \Leftrightarrow \forall x(x \notin B \to x \notin A) \Leftrightarrow \forall x(x \in \overline{B} \to x \in \overline{A}) \Leftrightarrow \overline{B} \subseteq \overline{A}$
23. 主修计算机科学而不主修数学，或主修数学而不主修计算机科学的所有学生的集合。
25. $(A \cup B) - (A \cap B)$的元素是属于 A 和 B 的并集却不属于 A 和 B 的交集的元素，也就是说它属于 A 或属于 B，但不同时属于 A 和 B。这正是元素属于 $A \oplus B$ 的含义。
27. a) $A \oplus A = (A - A) \cup (A - A) = \emptyset \cup \emptyset = \emptyset$
    b) $A \oplus \emptyset = (A - \emptyset) \cup (\emptyset - A) = A \cup \emptyset = A$
    c) $A \oplus U = (A - U) \cup (U - A) = \emptyset \cup \overline{A} = \overline{A}$
    d) $A \oplus \overline{A} = (A - \overline{A}) \cup (\overline{A} - A) = A \cup \overline{A} = U$
29. $B = \emptyset$
31. 是的。假定 $x \in A$ 但 $x \notin B$。如果 $x \in C$，那么 $x \notin A \oplus C$ 但 $x \in B \oplus C$，矛盾。如果 $x \notin C$，那么 $x \in A \oplus C$ 但 $x \notin B \oplus C$，矛盾。因此 $A \subseteq B$。类似地可知 $B \subseteq A$，从而 $A = B$。
33. 是的。
35. a) {1, 2, 3, …, n}　　b) {1}
37. a) A_n　　b) {0, 1}
39. a) {1, 2, 3, 4, 7, 8, 9, 10}
    b) {2, 4, 5, 6, 7}
    c) {1, 10}
41. 如果第一个位串的第 i 位是 1 而第二个位串的第 i 位为 0，则两个集合之差的位串的第 i 位是 1；否则为 0。
43. a) 11 1110 0000 0000 0000 0000 0000 ∨ 01 1100 1000 0000 0100 0101 0000 = 11 1110 1000 0000 0100 0101 0000，代表{a,b,c,d,e,g,p,t,v}
    b) 11 1110 0000 0000 0000 0000 0000 ∧ 01 1100 1000 0000 0100 0101 0000 = 01 1100 0000 0000 0000 0000 0000，代表{b,c,d}
    c) (11 1110 0000 0000 0000 0000 0000 ∨ 00 0110 0110 0001 1000 0110 0110) ∧ (01 1100 1000 0000 0100 0101 0000 ∨ 00 1010 0010 0000 1000 0010 0111)
    = 11 1110 0110 0001 1000 0110 0110 ∧ 01 1110 1010 0000 1100 0111 0111
    = 01 1110 0010 0000 1000 0110 0110，代表{b,c,d,e,i,o,t,u,x,y}
    d) 11 1110 0000 0000 0000 0000 0000 ∨ 01 1100 1000 0000 0100 0101 0000 ∨ 00 1010

0010 0000 1000 0010 0111 ∨ 00 0110 0110 0001 1000 0110 0110 = 11 1110 1110 0001 1100 0111 0111，代表 $\{a, b, c, d, e, g, h, i, n, o, p, t, u, v, x, y, z\}$

45. a) $\{1, 2, 3, \{1, 2, 3\}\}$ b) $\{\emptyset\}$
   c) $\{\emptyset, \{\emptyset\}\}$ d) $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$

47. a) $\{3\cdot a, 3\cdot b, 1\cdot c, 4\cdot d\}$
   b) $\{2\cdot a, 2\cdot b\}$
   c) $\{1\cdot a, 1\cdot c\}$
   d) $\{1\cdot b, 4\cdot d\}$
   e) $\{5\cdot a, 5\cdot b, 1\cdot c, 4\cdot d\}$

49. $\overline{F}$ = {0.4 Alice, 0.1 Brain, 0.6 Fred, 0.9 Oscar, 0.5 Rita}
   $\overline{R}$ = {0.6 Alice, 0.2 Brain, 0.8 Fred, 0.1 Oscar, 0.3 Rita}

51. $F \cap R$ = {0.4 Alice, 0.8 Brain, 0.2 Fred, 0.1 Oscar, 0.5 Rita}

1.6 节

1. a) $f(0)$无定义。
   b) 对 $x<0$，$f(x)$无定义。
   c) 对 x 的每个值都指派了两个值，$f(x)$不符合函数定义。

3. a) 不是函数。
   b) 是函数。
   c) 不是函数。

5. a) 整数集合。
   b) 非负偶数集合。
   c) 不超过 7 的非负整数集合。
   d) 整数平方的集合 = $\{0, 1, 4, 9, 16, \cdots\}$。

7. a) 1 b) 0 c) 0 d) −1
   e) 3 f) −1 g) 2 h) 1

9. 只有 a) 中的函数是。

11. 只有 a) 和 d) 中的函数是。

13. a) $x \geq 0$ 时函数 $f(x) = 3x + 1$，$x<0$ 时函数 $f(x) = -3x + 2$
   b) $f(x) = |x| + 1$
   c) $x \geq 0$ 时 $f(x) = 2x + 1$，$x<0$ 时 $f(x) = -2x$
   d) $f(x) = x^2 + 1$

15. a) 是 b) 不是 c) 是 d) 不是

17. a) $f(S) = \{0,1,3\}$
   b) $f(S) = \{0,1,3,5,8\}$
   c) $f(S) = \{0,8,16,40\}$
   d) $f(S) = \{1,12,33,65\}$

19. a) 令 x 和 y 为 A 中不同的元素。由于 g 是一对一的，$g(x)$和 $g(y)$是 B 中不同的元素。又因为 f 是一对一的，$f(g(x)) = (f\circ g)(x)$和 $f(g(y)) = (f\circ g)(y)$是 C 中不

同元素。因而 $f\circ g$ 是一对一的。

b) 令 $y\in C$。因为 f 是映上的，对某个 $b\in B$，$y=f(b)$。又因为 g 是映上的，对某个 $x\in A$，$b=g(x)$。因此 $y=f(b)=f(g(x))=(f\circ g)(x)$。由此知 $f\circ g$ 是映上的。

21. 不是。例如，假定 $A=\{a\}$，$B=\{b,c\}$，$C=\{d\}$。令 $g(a)=b$，$f(b)=d$，$f(c)=d$，于是 f 和 $f\circ g$ 为映上的，但 g 不是。

23. $(f+g)(x)=x^2+x+3$，$(fg)(x)=x^3+2x^2+x+2$

25. f 是一对一的，因为 $f(x_1)=f(x_2)\Leftrightarrow ax_1+b=ax_2+b\Leftrightarrow ax_1=ax_2\Leftrightarrow x_1=x_2$。$f$ 是映上的，因为 $f((y-b)/a)=y$，$f^{-1}(y)=(y-b)/a$。

27. 令 $f(1)=a$，$f(2)=a$，令 $S=\{1\}$，$T=\{2\}$。那么 $f(S\cap T)=f(\emptyset)=\emptyset$，但 $f(S)\cap f(T)=\{a\}\cap\{a\}=\{a\}$。

29. a) $\{x\mid 0\leqslant x<1\}$

b) $\{x\mid -1\leqslant x<2\}$

c) $\emptyset$

31. $f^{-1}(\overline{S})=\{x\in A\mid f(x)\notin S\}=\overline{\{x\in A\mid f(x)\in S\}}$

$=\overline{f^{-1}(S)}$。

33. 令 $x=\lfloor x\rfloor+\varepsilon$，其中 ε 为实数，$0\leqslant\varepsilon<1$。若 $\varepsilon<\frac{1}{2}$，那么 $\lfloor x\rfloor-1<x-\frac{1}{2}<\lfloor x\rfloor$，所以 $\lceil x-1/2\rceil=\lfloor x\rfloor$ 且这是最接近 x 的整数。若 $\varepsilon>\frac{1}{2}$，那么 $\lfloor x\rfloor<x-\frac{1}{2}<\lfloor x\rfloor+1$，所以 $\lceil x-1/2\rceil=\lfloor x\rfloor+1$ 且这是最接近 x 的整数。若 $\varepsilon=\frac{1}{2}$，那么 $\lceil x-1/2\rceil=\lfloor x\rfloor$，这是围绕 x 且到 x 的距离相等的两个整数中较小的一个。

35. 把实数 x 写成 $\lfloor x\rfloor+\varepsilon$，其中 ε 是实数，$0\leqslant\varepsilon<1$。因为 $\varepsilon=x-\lfloor x\rfloor$，所以 $0\leqslant-\lfloor x\rfloor<1$，由此直接得前两个不等式 $x-1<\lfloor x\rfloor$ 和 $\lfloor x\rfloor\leqslant x$。对另两个不等式，把 x 写成 $x=\lceil x\rceil-\varepsilon'$，其中 $0\leqslant\varepsilon'<1$。于是 $0\leqslant\lceil x\rceil-x<1$，从而立即得到想要的不等式。

37. a) 若 $x<n$，由于 $\lfloor x\rfloor\leqslant x$，所以 $\lfloor x\rfloor<n$。假定 $x\geqslant n$，由底函数的定义知 $\lfloor x\rfloor\geqslant n$。这表明如果 $\lfloor x\rfloor<n$，那么 $x<n$。

b) 若 $n<x$，那么由于 $x\leqslant\lceil x\rceil$，$n\leqslant\lceil x\rceil$。假定 $n\geqslant x$，由顶函数的定义知 $\lceil x\rceil\leqslant n$。这表明如果 $n<\lceil x\rceil$，那么 $n<\lceil x\rceil$。

39. 假定 $N\leqslant x<N+1$。若 $N+\frac{1}{2}\leqslant x$，那么 $\lfloor 2x\rfloor=2N+1$，$\lfloor x\rfloor=N$ 且 $\lfloor x+1/2\rfloor=N+1$，所以 $\lfloor 2x\rfloor=\lfloor x\rfloor+\lfloor x+1/2\rfloor$。如果 $x<N+\frac{1}{2}$，那么 $\lfloor 2x\rfloor=2N$，$\lfloor x\rfloor=\lfloor x+1/2\rfloor=N$ 等式仍然成立。

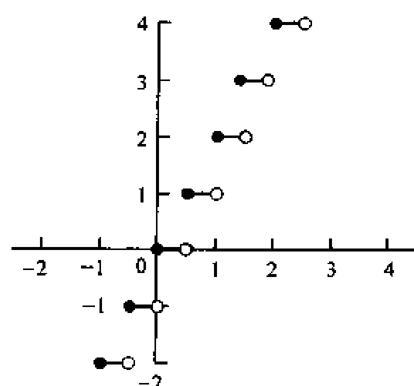
41. 假定 $x\geqslant 0$。等式左边是 $\lceil -x\rceil$ 而右边是 $-\lfloor x\rfloor$。如果 x 是整数，那么两边都等于 $-x$。否则令 $x=n+\varepsilon$，其中 n 为自然数，ε 为实数，$0\leqslant\varepsilon<1$。那么 $\lceil -x\rceil=\lceil -n-\varepsilon\rceil=-n$，且 $-\lfloor x\rfloor=-\lfloor n+\varepsilon\rfloor=-n$，等式成立。若 $x<0$，那么只要用 $-x$ 代替 x，即可得到想证明的等式。

43. $\lceil b\rceil-\lfloor a\rfloor-1$

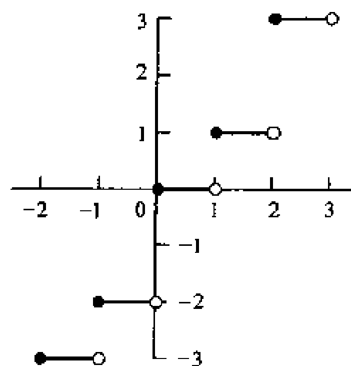
45. a) 1　　b) 3　　c) 126　　d) 3600

47. a) 100　　b) 256　　c) 1030　　d) 30 200

49.

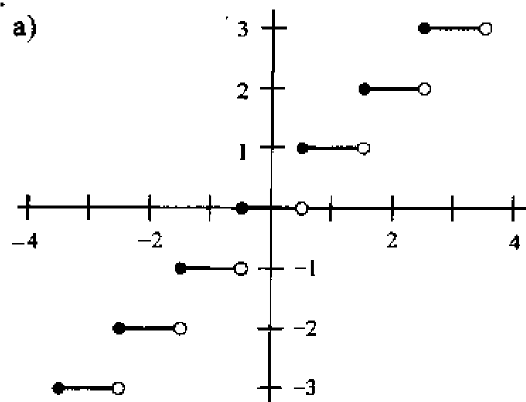


51.

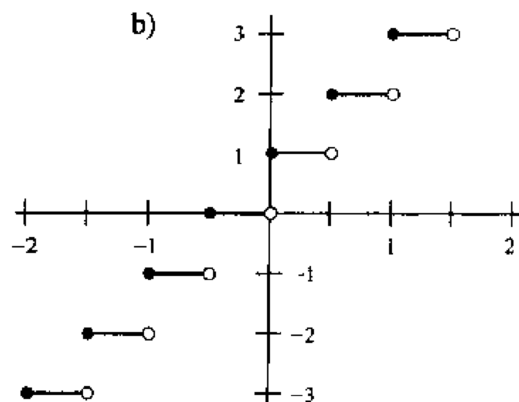


53.

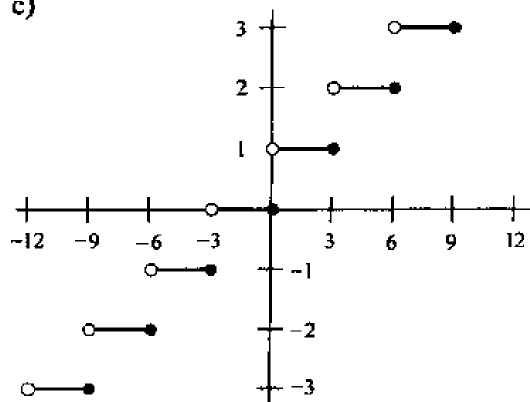
a)



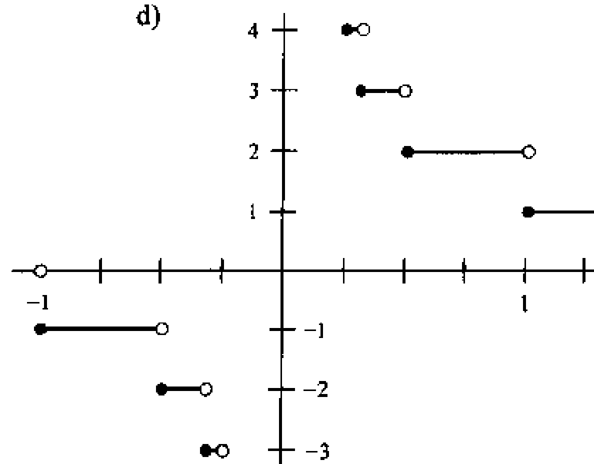
b)



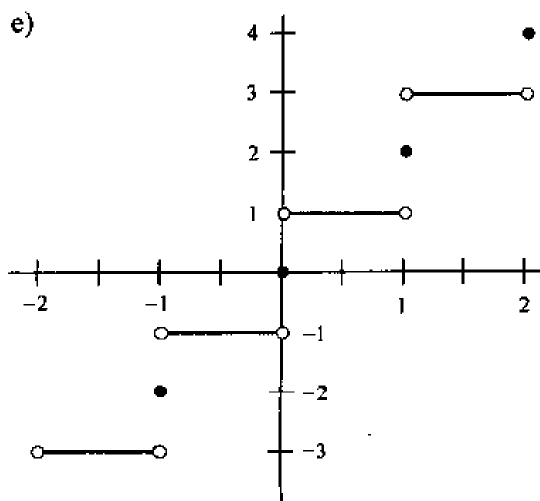
c)



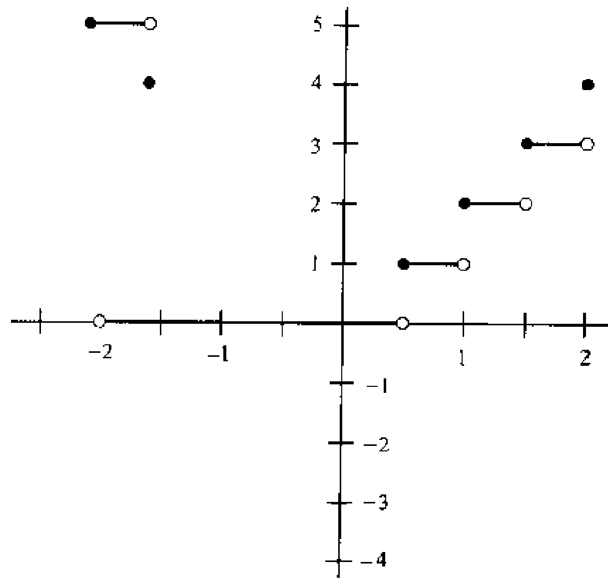
d)



e)



f)



55. $f^{-1}(y)=(y-1)^{1/3}$

57. a) $f_{A\cap B}(x)=1\Leftrightarrow x\in A\cap B\Leftrightarrow x\in A\wedge x\in B$
$\Leftrightarrow f_A(x)=1\wedge f_B(x)=1\Leftrightarrow f_A(x)f_B(x)=1$

b) $f_{A\cup B}(x)=1\Leftrightarrow x\in A\cup B\Leftrightarrow x\in A\vee x\in B$
$\Leftrightarrow f_A(x)=1\vee f_B(x)=1\Leftrightarrow f_A(x)+f_B(x)-f_A(x)f_B(x)=1$

c) $f_{\overline{A}}(x)=1\Leftrightarrow x\in\overline{A}\Leftrightarrow x\notin A\Leftrightarrow f_A(x)=0$
$\Leftrightarrow 1-f_A(x)=1$

d) $f_{A\oplus B}(x)=1\Leftrightarrow x\in A\oplus B\Leftrightarrow(x\in A\wedge x\notin B)\vee$
$(x\notin A\wedge x\in B)\Leftrightarrow f_A(x)+f_B(x)-2f_A(x)f_B(x)=1$

59. a) 定义域是 **Z**；伴域是 **R**；定义区域是非零整数集；f 的未定义区域是 $\{0\}$；不是全函数。

b) 定义域是 **Z**；伴域是 **R**；定义区域是 **Z**；f 的未定义区域是 $\emptyset$；全函数。

c) 定义域是 **Z**×**Z**；伴域是 **Q**；定义区域是**Z**×(**Z**−$\{0\}$)；f 的未定义区域是 $Z\times\{0\}$；不是全函数。

d) 定义域是 **Z**×**Z**；伴域是 **Z**；定义区域是 **Z**×**Z**；f 的未定义区域是 $\emptyset$；全函数。

e) 定义域是 **Z**×**Z**；伴域是 **Z**；定义区域是$\{(m,n)\mid m>n\}$；f 的未定义区域是$\{(m,n)\mid m\leqslant n\}$；不是全函数。

1.7 节

1. a) 3　　b) −1
c) 787　　d) 2639

3. a) $a_0=2$，$a_1=3$，$a_2=5$，$a_3=9$
b) $a_0=1$，$a_1=4$，$a_2=27$，$a_3=256$
c) $a_0=0$，$a_1=0$，$a_2=1$，$a_3=1$
d) $a_0=0$，$a_1=1$，$a_2=2$，$a_3=3$

5. a) 2，5，8，11，14，17，20，23，26，29
   b) 1，1，1，2，2，2，3，3，3，4
   c) 1，1，3，3，5，5，7，7，9，9
   d) −1，−2，−2，8，88，656，4912，40064，362368，3627776
   e) 3，6，12，24，48，96，192，384，768，1536
   f) 1，1，2，3，5，8，13，21，34，55
   g) 1，2，2，3，3，3，3，4，4，4
   h) 3，3，5，4，4，3，5，5，4，3
7. 每一项可以是前一项的两倍；第 n 项可以从前一项加上 $n-1$ 得到；这些项可以是并非 3 的倍数的正整数；还有无穷多种其他的可能性。
9. a) 一个 1 和一个 0，再跟上二个 1 和二个 0，再跟上三个 1 和三个 0，等等。
   b) 正整数按增序排列，而且每个偶数出现两次。
   c) 奇数位的项是 2 的连续性幂，偶数位的项全是 0。
   d) $a_n = 3\cdot 2^{n-1}$
   e) $a_n = 15 - 7(n-1) = 22 - 7n$
   f) $a_n = (n^2 + n + 4)/2$
   g) $a_n = 2n^3$
   h) $a_n = n! + 1$
11. 由于 a_n 表示第 n 个不是完全平方的正整数，在 1，2，…，a_n 中，a_1，a_2，…，a_n 是那些不为完全平方的数，1^2，2^2，…，k^2 是为完全平方的数，其中 k 是使 $k^2 < n + k < (k+1)^2$ 的整数。于是 $a_n = n + k$，其中 $k^2 < a_n < (k+1)^2$。为求 k，首先注意 $k^2 < n + k < (k+1)^2$，所以，$k^2 + 1 \leqslant n + k \leqslant (k+1)^2 - 1$。因而 $(k - \frac{1}{2})^2 + \frac{3}{4} = k^2 - k + 1 \leqslant n \leqslant k^2 + k = (k + \frac{1}{2})^2 - \frac{1}{4}$。由此可知 $k - \frac{1}{2} < \sqrt{n} < k + \frac{1}{2}$，所以 $k = \{\sqrt{n}\}$，$a_n = n + k = n + \{\sqrt{n}\}$。
13. a) 20　b) 11　c) 30　d) 511
15. a) 1533　b) 510　c) 4923　d) 9842
17. a) 21　b) 78　c) 18　d) 18
19. $\sum_{j=1}^{n}(a_j - a_{j-1}) = a_n - a_0$
21. a) n^2　b) $n(n+1)/2$
23. 15150
25. $\frac{n(n+1)(2n+1)}{3} + \frac{n(n+1)}{2} + (n+1)(m-(n+1)^2+1)$，其中 $n = \lfloor\sqrt{m}\rfloor - 1$
27. a) 0　b) 1680
    c) 1　d) 1024
29. 34
31. a) 可数，−1，−2，−3，−4，…
    b) 可数，0，2，−2，4，−4，…
    c) 不可数。

d) 可数，0，7，−7，14，−14，…

33. 假定 $A-B$ 可数，那么由于 $A=(A-B)\cup B$，A 中元素可以用交替列出 $A-B$ 中元素和 B 中元素的方式排成序列。这和 A 不可数矛盾。

35. 假定 B 可数，那么 B 中元素可以列成 b_1，b_2，b_3，…。由于 A 是 B 的子集，取 $\{b_n\}$ 序列的子序列使它包含属于 A 的项，这样就得到 A 中元素的一个序列。但 A 是不可数的，不可能排成序列。

37. 假定 A_1，A_2，A_3，…全是可数集合。由于 A_i 是可数的，可以把它的元素排成序列 a_{i1}，a_{i2}，a_{i3}，…。集合 $\bigcup_{i=1}^{n} A_i$ 的元素 a_{ij} 可以按下列方式排成序列：先排 $i+j=2$ 的元素，再排 $i+j=3$ 的元素，然后是 $i+j=4$ 的元素，等等。

39. 长度为 m 的位串只有有限多个，即 2^m 个。所有位串的集合是长度为 m 的位串的集合的并集，$m=0$，1，2，…由于可数的多个可数集合的并集是可数的，所以有可数的多个位串。

41. 对任何有限的字母表，只有有限多个长度为 n 的位串，其中 n 为正整数。从练习 37 可知，有限字母表上长度不超过给定正整数的位串只有有限多个。由于用特定程序语言写的计算机程序集合是某个有限字母表上的所有位串之集合的子集，而根据练习 34，所有位串的集合是可数的，所以作为子集的程序集合是可数的。

43. 练习 41 证明了只有可数的多个计算机程序，因此只有可数的多个可计算的函数。根据练习 42 所示，函数个数是不可数的，所以并非所有函数都是可计算的。

1.8 节

1. a) 是　b) 是　c) 不是　d) 是　e) 是　f) 是

3. 对所有 $x>9$，$x^4+9x^3+4x+7\leqslant 4x^4$，所以 x^4+9x^3+4x+7 是 $O(x^4)$。

5. 对所有 $x>1$，$(x^2+1)/(x+1)=x-1+2/(x+1)<x$，因此 $(x^2+1)/(x+1)$ 是 $O(x)$。

7. a) 3　b) 3　c) 1　d) 0

9. 对所有 $x>17$，$x^2+4x+17\leqslant 3x^3$，所以 $x^2+4x+17$ 是 $O(x^3)$。可是若 x^3 是 $O(x^2+4x+17)$，那么对某个 C 和足够大的 x，$x^3<C(x^2+4x+17)<3Cx^2$。这表明对足够大的 x，$x<C$，而这是不可能的，所以 x^3 不是 $O(x^2+4x+17)$。

11. 对 $x>1$，$3x^4+1\leqslant 4x^4=8(x^4/2)$，所以 $3x^4+1$ 是 $O(x^4/2)$。另一方面，对 $x>0$，$x^4/2\leqslant 3x^4+1$，所以 $x^4/2$ 是 $O(3x^4+1)$。

13. 因为对所有 $n>0$，$2^n<3^n$，所以 2^n 是 $O(3^n)$。但如果 3^n 也是 $O(2^n)$，那么对某个 C，$3^n\leqslant C\cdot 2^n$ 对足够大的 n 应成立。这表示 $C\geqslant(3/2)^n$ 对所有足够大的 n 应成立，可这是不可能的。所以 3^n 不是 $O(2^n)$。

15. 对函数 $f(x)$ 而言，必有实数 k 和 C，使 $x>k$ 时 $|f(x)|\leqslant C$。因此这是那些对足够大的 x 有定界的函数。

17. 由已知条件知有常数 C_1，C_2，k_1 和 k_2，使得对所有 $x>k_1$，$|f(x)|\leqslant C_1|g(x)|$；对所有 $x>k_2$，$|g(x)|\leqslant C_2|h(x)|$。因此对 $x>\max(k_1,k_2)$，$|f(x)|\leqslant C_1|g(x)|\leqslant C_1C_2|h(x)|$。这说明 $f(x)$ 是 $O(h(x))$。

19. a) $O(n^3)$　b) $O(n^5)$　c) $O(n^3\cdot n!)$

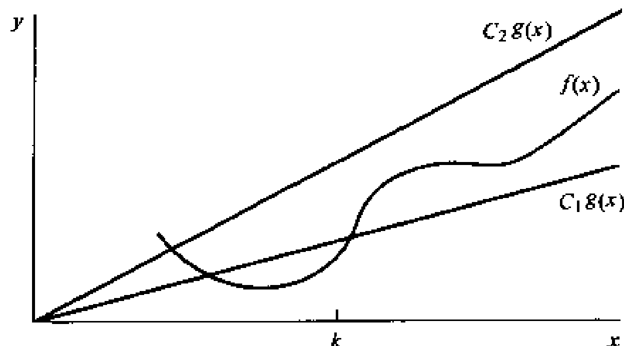
21. a) $O(n^2\log n)$ b) $O(n^2(\log n)^2)$ c) $O(n^{2^n})$

23. a) 既非$\Theta(x^2)$，也非$\Omega(x^2)$。
    b) $\Theta(x^2)$，和$\Omega(x^2)$。
    c) 既非$\Theta(x^2)$，也非$\Omega(x^2)$。
    d) $\Omega(x^2)$，但非$\Theta(x^2)$。
    e) $\Omega(x^2)$，但非$\Theta(x^2)$。
    f) $\Omega(x^2)$和$\Theta(x^2)$。

25. 如果 $f(x)$是$\Theta(g(x))$，那么存在常数 C_1，C_2，使 $C_1|g(x)|\leqslant|f(x)|\leqslant C_2|g(x)|$。于是$|f(x)|\leqslant C_2|g(x)|$，$|g(x)|\leqslant(1/C_1)|f(x)|$对 $x>k$ 成立。因此，$f(x)$是 $O(g(x))$，$g(x)$是 $O(f(x))$。反过来，假定$f(x)$是 $O(g(x))$，$g(x)$是 $O(f(x))$，那么存在常数 C_1，C_2，k_1 和 k_2，使得对于 $x>k_1$，$|f(x)|\leqslant C_1|g(x)|$；对于 $x>k_2$，$|g(x)|\leqslant C_2|f(x)|$。假定 $C_2>0$（我们总可以增大 C_2），于是对于 $x>\max(k_1,k_2)$，$(1/C_2)|g(x)|\leqslant|f(x)|\leqslant C_1|g(x)|$。所以，$f(x)$是 $\Theta(g(x))$。

27. 如果 $f(x)$是 $\Theta(g(x))$，那么$f(x)$既是$O(g(x))$，又是$\Omega(g(x))$。因此存在常数 C_1，C_2，k_1，k_2，使得对于 $x>k_2$，$|f(x)|\leqslant C_2|g(x)|$；对于 $x>k_1$，$|f(x)|\geqslant C_1|g(x)|$。于是只要 $x>k$，其中 $k=\max(k_1,k_2)$，就有 $C_1|g(x)|\leqslant|f(x)|\leqslant C_2|g(x)|$。反过来，如果有正整 C_1，C_2 和 k，使 $C_1|g(x)|\leqslant|f(x)|\leqslant C_2|g(x)|$对所有 $x>k$ 成立，那么令 $k_1=k_2=k$，可知 $f(x)$既是 $O(g(x))$，又是 $\Theta(g(x))$。

29.



31. 如果 $f(x)$是 $\Theta(1)$，那么$|f(x)|$限于正常数 C_1 和 C_2 之间。换言之，$f(x)$不可能增大到超过一个固定的界，也不能小于这个界的负数，而且必定不会比某个固定的界更接近于0。

33. 由于 $f(x)$是 $O(g(x))$，所以一定有常数 C 和 1，使得对于所有 $x>1$，$|f(x)|\leqslant C|g(x)|$。于是$|f^k(x)|\leqslant C^k|g^k(x)|$对所有 $x>1$ 成立。所以取常数为 C^k，$f^k(x)$是 $O(g^k(x))$。

35. 由于 $f(x)$和 $g(x)$均是增长的且有界，我们可以假设对足够大的 x，$f(x)\geqslant1$ 且$g(x)\geqslant1$。有常数 C 和k，使 $x>k$ 时$f(x)\leqslant Cg(x)$。这表明对足够大的 x，$\log f(x)\leqslant\log C+\log g(x)<2\log g(x)$。因此 $\log f(x)$ 是 $O(\log g(x))$。

37. 根据定义，有正常数 C_1，C'_1，C_2，C'_2，k_1，k'_1，k_2 和 k'_2，使得对所有 $x>k_1$，$f_1(x)\geqslant C_1|g(x)|$；对所有 $x>k'_1$，$f_1(x)\leqslant C'_1|g(x)|$；对所有 $x>k_2$，$f_2(x)\geqslant C_2|g$

$(x)|$；对所有 $x>k'_2$，$f_2(x)\leqslant C'_2|g(x)|$。第1个和第3个不等式相加，知 $f_1(x)+f_2(x)\geqslant(C_1+C_2)|g(x)|$ 对所有 $x>k$ 成立，其中 $k=\max(k_1,k_2)$；第2个和第4个不等式相加得 $f_1(x)+f_2(x)\leqslant(C'_1+C'_2)|g(x)|$ 对所有 $x>k'$ 成立，其中 $k'=\max(k'_1,k'_2)$。可见 $f_1(x)+f_2(x)$ 是 $\Theta(g(x))$。如果 f_1 和 f_2 能取负值，这一结论就不一定对了。

39. 结论不成立。令 $f_1(x)=x^2+2x$，$f_2(x)=x^2+x$，$g(x)=x^2$。于是 $f_1(x)$ 和 $f_2(x)$ 都是 $O(g(x))$，但 $(f_1-f_2)(x)$ 不是。

41. $f(n)$ 是这样的函数，如果 n 是奇正数，$f(n)=n$；如果 n 为偶正数，$f(n)=1$。$g(n)$ 相反，如果 n 是奇正数，$g(n)=1$；如果 n 为偶正数，$g(n)=n$

43. 有正常数 C_1，C'_1，C_2，C'_2，k_1，k'_1，k_2 和 k'_2，使得对所有 $x>k_1$，$|f_1(x)|\geqslant C_1|g_1(x)|$；对所有 $x\geqslant k'_1$，$|f_1(x)|\leqslant C'_1|g_1(x)|$；对所有 $x>k_2$，$|f_2(x)|>C_2|g_2(x)|$；对所有 $x>k'_2$，$|f_2(x)|\leqslant C'_2|g_2(x)|$。由于 f_2 和 g_2 总不为0，最后两个不等式可以重写为对所有 $x>k_2$，$|1/f_2(x)|\leqslant(1/C_2)|1/g_2(x)|$；对所有 $x>k'_2$，$|1/f_2(x)|\geqslant(1/C'_2)|1/g_2(x)|$。把第1个不等式和重写过的第4个不等式相乘可得知，对所有 $x>\max(k_1,k'_2)$，$|f_1(x)/f_2(x)|\geqslant(C_1/C'_2)|g_1(x)/g_2(x)|$；把第2个不等式和重写过的第3个不等式相乘可得知，对所有 $x>\max(k_2,k'_1)$，$|f_1(x)/f_2(x)|\leqslant(C'_1/C_2)\cdot|g_1(x)/g_2(x)|$。因此 f_1/f_2 是 $\Theta(g_1/g_2)$。

45. 有正常数 C_1，C_2，k_1，k_2，k'_1，k'_2 使得对所有 $x>k_1$，$y>k_2$，$|f(x,y)|\leqslant C_1|g(x,y)|$；对所有 $x>k'_1$，$y>k'_2$，$|f(x,y)\geqslant C_2|g(x,y)|$。

47. 对 $x>1$ 和 $y>1$，$(x^2+xy+x\log y)^3<(3x^2y^3)=27x^6y^3$，这是因为 $x^2<x^2y$，$xy<x^2y$ 且 $x\log y<x^2y$。因此 $(x^2+xy+x\log y)^3$ 是 $O(x^6y^3)$。

49. 对所有正实数 x 和 y，$\lfloor xy\rfloor\leqslant xy$。因此，由定义知 $\lfloor xy\rfloor$ 是 $O(xy)$，只要取 $C=1$，$k_1=k_2=0$。

51. a) $\lim\limits_{x\to\infty}x^2/x^3=\lim\limits_{x\to\infty}1/x=0$

b) $\lim\limits_{x\to\infty}\dfrac{x\log x}{x^2}=\lim\limits_{x\to\infty}\dfrac{\log x}{x}=\lim\limits_{x\to\infty}\dfrac{1}{x\ln 2}=0$

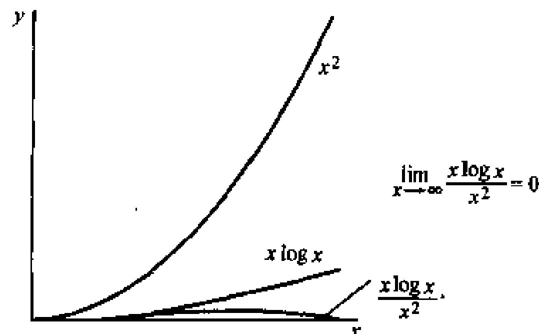
(使用 L'Hopital 规则)

c) $\lim\limits_{x\to\infty}\dfrac{x^2}{2^x}=\lim\limits_{x\to\infty}\dfrac{2x}{2^x\cdot\ln 2}=\lim\limits_{x\to\infty}\dfrac{2}{2^x\cdot(\ln 2)^2}=0$

(使用 L'Hopital 规则)

d) $\lim\limits_{x\to\infty}\dfrac{x^2+x+1}{x^2}=\lim\limits_{x\to\infty}(1+\dfrac{1}{x}+\dfrac{1}{x^2})=1\neq 0$

53.



55. 不是。取 $f(x)=1/x^2$ 和 $g(x)=1/x$。

57. a) 由于 $\lim_{x\to\infty} f(x)/g(x)=0$，$|f(x)|/|g(x)|<1$ 对足够大的 x 成立，因此 $|f(x)|<|g(x)|$ 对 $x>k$ 成立，其中 k 是某个常数。于是 $f(x)$ 是 $O(g(x))$。

   b) 令 $f(x)=g(x)=x$。那么 $f(x)$ 是 $O(g(x))$，但 $f(x)$ 不是 $o(g(x))$，因为 $f(x)/g(x)=1$。

59. 因为 $f_2(x)$ 是 $o(g(x))$，由练习 57 a)可知 $f_2(x)$ 是 $O(g(x))$。由推论 1 可知 $f_1(x)+f_2(x)$ 是 $O(g(x))$。

61. 易于证明：对 $i=0, 1, \cdots, n-1$，$(n-i)(i+1)\geqslant n$。因此 $(n!)^2=(n\cdot 1)((n-1)\cdot 2)((n-2).3)\cdots(2\cdot(n-1))(1\cdot n)\geqslant n^n$，因此，$2\log n!\geqslant n\log n$。

63. a) 渐近　b) 不渐近　c) 渐近　d) 渐近
   e) 不渐近　f) 不渐近　g) 不渐近

补充练习

1. a) $q\to p$　b) $q\wedge p$　c) $\neg q\vee\neg p$　d) $q\leftrightarrow p$

3. a) 除非 $\neg p$ 为假，即 p 为真，否则命题不会为假。如果 p 为真，q 为真，那么 $\neg q\wedge(p\to q)$ 为假，蕴含为真。如果 p 为真而 q 为假，那么 $p\to q$ 为假，所以 $\neg q\wedge(p\to q)$ 仍为假，蕴含还是为真。

   b) 除非 q 为假，否则命题不会为假。如果 q 为假，p 为真，那么 $(p\vee q)\wedge\neg p$ 为假，蕴含为真。若 q 为假，p 为假，那么 $(p\vee q)\wedge\neg p$ 也为假，蕴含仍为真。

5. $(p\wedge q\wedge r\wedge\neg s)\vee(p\wedge q\wedge\neg r\wedge s)\vee(p\wedge\neg q\wedge r\wedge s)\vee(\neg p\wedge q\wedge r\wedge s)$

7. a) F　b) T　c) F　d) T　e) F　f) T

9. 假定 $\exists x(p(x)\to Q(x))$ 为真。那么或者对某个 x_0，$Q(x_0)$ 为真，这时 $\forall xP(x)\to\exists xQ(x)$ 为真；或者有某个 x_0，$P(x_0)$ 为假，这时 $\forall xP(x)\to\exists xQ(x)$ 也为真。反过来，假定 $\exists x(P(x))\to Q(x))$ 为假。这表明 $\forall x(P(x)\wedge\neg Q(x))$ 为真，即 $\forall xP(x)$ 为真，$\forall x(\neg Q(x))$ 也为真。后一命题等价于 $\neg\exists xQ(x)$，因此 $\forall xP(x)\to\exists xQ(x)$ 为假。

11. 不。

13. $\forall x\forall z\exists yT(x,y,z)$，其中 $T(x,y,z)$ 为“学生 x 已选修 z 系的 y 课”，相应的论域是班上所有学生的集合，这所大学开设的所有课程的集合，以及数学学院所有系的集合。

15. a) $\overline{A}$　b) $A\cap B$　c) $A-B$　d) $\overline{A}\cap\overline{B}$　e) $A\oplus B$

17. 是

19. a) $A\cap\overline{A}=\{x\mid x\in A\wedge x\notin A\}=\varnothing$
   b) $A\cup\overline{A}=\{x\mid x\in A\vee x\notin A\}=U$

21. $A-(A-B)=A-(A\cap\overline{B})=A\cap(\overline{A\cap\overline{B}})=A\cap(\overline{A}\cup B)$
$=(A\cap\overline{A})\cup(A\cap B)=\varnothing\cup(A\cap B)=A\cap B$

23. 令 $A=\{1\}$，$B=\varnothing$，$C=\{1\}$，于是 $(A-B)-C=\varnothing$，但 $A-(B-C)=\{1\}$。

25. 不等。例如，令 $A=B=\{a, b\}$，$C=\varnothing$，$D=\{a\}$。于是 $(A-B)-(C-D)=\varnothing-\varnothing=\varnothing$，但 $(A-C)-(B-D)=\{a,b\}-\{b\}=\{a\}$。

27. a) $|\varnothing|\leqslant|A\cap B|\leqslant|A|\leqslant|A\cup B|\leqslant|U|$

b) $|\varnothing| \leqslant |A-B| \leqslant |A \oplus B| \leqslant |A \cup B| \leqslant |A|+|B|$

29. a) f 是，g 不是。

b) f 是，g 不是。

c) f 有逆：$f^{-1}(a)=3$，$f^{-1}(b)=4$，$f^{-1}(c)=2$，$f^{-1}(d)=1$；g 没有逆。

31. 令 $f(a)=f(b)=1$，$f(c)=f(d)=2$，$S=\{a,c\}$，$T=\{b,d\}$。于是 $f(S\cap T)=f(\varnothing)=\varnothing$，但 $f(S)\cap f(T)=\{1,2\}\cap\{1,2\}=\{1,2\}$。

33. a) 60　b) 6144　c) 20　d) 0

35. 令 p_n 为至多 n 阶的多项式集合，多项式以绝对值不超过 n 的整数为系数。那么对所有 n，p_n 是有限的。因为每个 n 阶多项式至多有 n 个不同的实根，所以只有有限多个代数数，它们是 p_n 中多项式的根。由于代数数是 p_n 中多项式的根集合的并集，$n=1, 2, 3, \ldots$，根据 1.7 节练习 37 可知，这是个可数集合。

37. $O(x^2 2^x)$

39. 注意 $\frac{n!}{2^n}=\frac{n}{2}\cdot\frac{n-1}{2}\cdots\frac{3}{2}\cdot\frac{2}{2}\cdot\frac{1}{2}>\frac{n}{2}\cdot 1\cdot 1\cdots 1\cdot\frac{1}{2}=\frac{n}{4}$。由于在 n 增长时 $\frac{n!}{2^n}$ 无限增长，$n!$ 在 n 足够大时不可能以某个常数与 2^n 之积为界，所以 $n!$ 不是 $O(2^n)$。

第 2 章

2.1 节

1. $max := 1$，$i := 2$，$max := 8$，$i := 3$，$max := 12$，$i := 4$，$i := 5$，$i := 6$，$i := 7$，$max := 14$，$i := 8$，$i := 9$，$i := 10$，$i := 11$

3. **Procedure** *sum* $(a^1\cdots, a_n$: 整数)

```
sum := a_1
for i := 2 to n
    sum := sum + a_i
{sum 为想要的值}
```

5. **Procedure** *interchange* $(x, y$: 实数)

```
z := x
x := y
y := z
```

需要赋值的最小数量是 3 个。

7. 线性搜索：$i := 1$，$i := 2$，$i := 3$，$i := 4$，$i := 5$，$i := 6$，$i := 7$，*location* $:= 7$；对分搜索：$i := 1$，$j := 8$，$m := 4$，$i := 5$，$m := 6$，$i := 7$，$m := 7$，$j := 7$，*location* $:= 7$

9. **Procedure** *insert* $(x, a_1, a_2, \cdots, a_n$: 整数)

```
        {列表的次序：a_1≤a_2≤…≤a_n}
a_{n+1} := x + 1
i := 1
while x > a_i
```

```
    i := i + 1
  for j := 0 to n - i
    a_{n-j+1} := a_{n-j}
    a_i := x
  {x 已被插入到正确位置}
```

11. **Procedure** *first largest* (a_1, …, a_n: 整数)

```
  max := a_1
  location := 1
  for i := 2 to n
  begin
    if max < a_i then
      begin
        max := a_i
        location := i
      end
  end
```

13. **Procedure** *mean − median − max − min* (a, b, c: 整数)

```
  mean := (a + b + c)/3
  关于≥的 a, b, c 的 6 个不同的顺序将会单独处理。
  if a > b then
    begin
      if b > c then
        median := b; max := a; min := c
    end
  算法的其余部分类似
```

15. **Procedure** *first − three* (a_1, a_2, …, a_n: 整数)

```
  if a_1 > a_2 then 交换 a_1 和 a_2
  if a_2 > a_3 then 交换 a_2 和 a_3
  if a_1 > a_2 then 交换 a_1 和 a_2
```

17. **Procedure** *onto* (f: 从 A 到 B 的函数，其中 $A = \{a_1, \cdots, a_n\}$, $B = \{b_1, \cdots, b_m\}$, a_1, …, a_n, b_1, …, b_m 是整数)

```
  for i := 1 to m
    hit (b_i) := 0
    count := 0
  for j := 1 to n
    if hit (f(a_j)) = 0 then
      begin
        hit (f(a_j)) := 1
        count := count + 1
```

```
    end
  if count = m then onto := true
  else onto := false
19. Procedure ones (a: 位串, a = a1a2…an)
  ones := 0
  for i := 1 to n
    begin
      if ai := 1 then
        ones := ones + 1
    end {ones 是位串 a 中 ones 的个数}
21. Procedure ternary search (s: 整数, a1, a2, …, an: 递增的整数)
  i := 1
  j := n
  while i < j - 1
    begin
      l = ⌊(i + j)/3⌋
      u = ⌊2(i + j)/3⌋
      if x > au then i := u + l
       elseif x > a1 then
         begin
           i := l + 1
           j := u
         end
       else j := l
    end
  if x = ai then location := i
  else location := 0
  {location 是等于 x 项的下标 0 没有找到}
23. Procedure find a mode (a1, a2, …, an: 非递减的整数)
  modecount := 0
  i := 1
  while i ≤ n
    begin
      value := ai
      count := 1
      while i ≤ n 和 ai = value
        begin
          count := count + 1
          i := i + 1
```

```
        end
      if count > modecount then
        begin
            modecount： count
            mode := value
        end
    end
  {mode 是最常出现的第 1 个值}
25. Procedure find duplicate (a_1, a_2, …, a_n 整数)
  location := 0
  i := 2
  while i ≤ n 和 location = 0
    begin
      j := 1
      while j < i 和 location = 0
      if a_i = a_j then location := i
      else j := j + 1
      i := i + 1
    end
  {location 是序列中重复先前值的第 1 个值的下标}
27. Procedure find decrease (a_1, a_2, …, a_n: 正整数)
  location := 0
  i := 2
  while i ≤ n 和 location = 0
      if a_i < a_{i-1} then location := i
      else i := i + 1
  {location 是值小于其前面那个数的第 1 个值的下标}
```

2.2 节

1. $2n-1$
3. 线性
5. $O(n)$
7. a) *power* := 1，y := 1；i := 1，*power* := 2，y := 3；i := 2，*power* := 4，y := 15
   b) $2n$ 次乘法和 n 次加法。
9. a) $2^{10^9} \sim 10^{3\times 10^8}$　b) 10^9　c) 3.96×10^7
   d) 3.16×10^4　e) 29　f) 12
11. a) 36 年　b) 13 天　c) 19 分钟
13. 平均比较次数是 $(3n+4)/2$。
15. $O(\log n)$

17. $O(n)$
19. $O(n^2)$
21. $O(n)$

2.3 节

1. a) 能 b) 不能 c) 能 d) 不能
3. 假设 $a|b$，那么必有整数 k，使 $ka=b$。由 $a(ck)=bc$ 知 $a|bc$。
5. 假定 $a|b$ 和 $b|a$，那么有整数 c 和 d，使 $b=ac$ 和 $a=bd$。因此 $a=acd$。由于 $a\neq0$，所以 $cd=1$。因而或 $c=d=1$，或 $c=d=-1$，也就是 $a=b$ 或 $a=-b$
7. 由于 $ac|bc$，必有整数 k，使 $ack=bc$，从而 $ak=b$，于是 $a|b$。
9. a) 2，5 b) −11，10 c) 34，7 d) 77，0
   e) 0，0 f) 0，3 g) −1，2 h) 4，0
11. $2^8\cdot3^4\cdot5^2\cdot7$
13. 假定 $log_2 3=a/b$，其中 a，$b\in\mathbf{Z}^+$ 且 $b\neq0$。于是 $2^{a/b}=3$，从而 $2^a=3^b$。这不合算术基本定理。因此 $log_2 3$ 是无理数。
15. a) 是 b) 不是 c) 是 d) 是
17. 如果 a mod $m=b$ mod m，那么 a 和 b 被 m 除时余数相同。于是 $a=q_1m+r$，$b=q_2m+r$，其中 $0\leqslant r<m$。由此得 $a-b=(q_1-q_2)m$，所以 $m|(a-b)$。由此得 $a\equiv b$ (mod m)。
19. 假定 n 不是素数，于是 $n=ab$，其中 a 和 b 为大于 1 的整数。由于 $a>1$，根据提示，2^a-1 是 2^n-1 的一个大于 1 的因子，2^n-1 的另一个因子也大于 1，所以 2^n-1 不是素数。
21. a) 2 b) 4 c) 12
23. $\phi(p^k)=p^k-p^{k-1}$
25. 有某个 b 使 $(b-1)k<n\leqslant bk$，因而 $(b-1)k\leqslant n-1<bk$。两边除以 k 得 $b-1<n/k\leqslant b$ 和 $b-1\leqslant(n-1)/k<b$。因此 $\lceil n/k\rceil=b$ 且 $\lfloor(n-1)/k\rfloor=b-1$。
27. 如果 x mod $m\leqslant\lceil m/2\rceil$，则 x mod m；如果 x mod $m>\lceil m/2\rceil$，则 $(x$ mod $m)-m$。
29. a) 1 b) 2 c) 3 d) 9
31. a) 否 b) 否 c) 是 d) 否
33. 由于 $\min(x,y)+\max(x,y)=x+y$，p_i 在 $\gcd(a,b)\cdot\text{lcm}(a,b)$ 的素数分解中的指数是 p_i 在 a 和 b 的素数分解中指数的和。
35. 令 $m=tn$。由于 $a\equiv b$ (mod m)，必有整数 s 使 $a=b+sm$。因此 $a=b+(st)n$，所以 $a\equiv b$(mod n)
37. 令 $m=c=2$，$a=0$，$b=1$，于是 $0=ac\equiv bc=2$ (mod 2)，但 $0=a\not\equiv b=1$ (mod 2)。
39. 由于 $a\equiv b$ (mod m)，必有整数 s 使 $a=b+sm$，所以 $a-b=sm$。于是 $k\geqslant2$ 时，$a^k-b^k=(a-b)(a^{k-1}+a^{k-2}b+\cdots+ab^{k-2}+b^{k-1}$ 也是 m 的倍数。于是 $a^k\equiv b^k$(mod m)。
41. a) 7，19，7，7，18，0
    b) 取下一可用空间 mod 31。
43. 2，6，7，10，8，2，6，7，10，8，…

45. a) GR QRW SDVV JR
    b) QB ABG CNFF TB
    c) QX UXM AHJJ ZX
47. 0
49. 本书的 ISBN 检验位是有效的，因为 $1\cdot0+2\cdot0+3\cdot7+4\cdot2+5\cdot8+6\cdot9+7\cdot9+8\cdot0+9\cdot5+10\cdot0\equiv0 \pmod{11}$。
51. a) 如果 n 为素数，$a_n=1$；否则 $a_n=0$。
    b) a_n 是 n 的最小素因子，且 $a_1=1$。
    c) a_n 是 n 的正除数的个数。
    d) 如果 n 没有大于 1 的完全平方正除数，$a_n=1$；否则 $a_n=0$。
    e) a_n 是小于或等于 n 的最大素数。
    f) a_n 是前 $n-1$ 个素数的乘积。

2.4 节

1. a) 6 b) 3 c) 11 d) 3
3. 8
5. a) 1110 0111 b) 1 0001 1011 0100
   c) 1 0111 1101 0110 1100
7. a) 31 b) 513 c) 341 d) 26 896
9. 将每六个数字转换为四个字位。
11. a) 1000 0000 1110
    b) 1 0011 0101 1010 1011
    c) 1010 1011 1011 1010
    d) 1101 1110 1111 1010 1100 1110 1101
13. 整数的二进展开是唯一的这种和。
15. 令 $a=(a_{n-1}a_{n-2}\cdots a_1a_0)_{10}$，那么 $a=10^{n-1}a_{n-1}+10^{n-2}a_{n-2}+\cdots+10a_1+a_0\equiv a_{n-1}+a_{n-2}+\cdots+a_1+a_0 \pmod 3$，这是因为对所有非负整数 j，$10^j\equiv1 \pmod 3$。于是 $3\mid a$ 当且仅当 3 能除尽 a 的十进数字之和。
17. 令 $a=(a_{n-1}a_{n-2}\cdots a_1a_0)_2$，那么 $a=a_0+2a_1+2^2a_2+\cdots+2^{n-1}a_{n-1}\equiv a_0-a_1+a_2-a_3+\cdots\pm a_{n-1} \pmod 3$。于是 a 能被 3 整除当且仅当偶数位的二进数字之和减去奇数位的二进数字之和能被 3 整除。
19. a) -6 b) 13 c) -14 d) 0
21. 和的 1 补表示可以这样计算：两个整数的 1 补相加，不过首位的进位要用做和的末位的进位。
23. 若 $m\geqslant0$，那么 m 的 1 补展开的首位 a_{n-1} 为 0，公式为 $m=\sum_{i=0}^{n-2}a_i2^i$。这一公式正确，因为右边正是 m 的二进展开。若 m 是负的，那么 m 的 1 补展开的首位 a_{n-1} 是 1。其他的 $n-1$ 位可以用 $111\ldots1$（共 $n-1$ 个 1）减去 $-m$ 得到，因为 1 减去一个字位就是该字位的补。于是位串 $a_{n-2}a_{n-3}\cdots a_1a_0$ 恰是 $(2^{n-1}-1)-(-m)$ 的二进展开。解方程 $(2^{n-1}-1)-(-m)=\sum_{i=0}^{n-2}a_i2^i$ 求 m 即可得想证明的等式，因为 $a_{n-1}=1$。

25. a) −7　b) 13　c) −15　d) −1

27. 要得到两个整数之和的 2 补表示，把它们的 2 补表示相加（就像加两个二进整数一样），忽略最左边一列的进位。不过若出现溢出，则结果就不正确了。发生溢出的情况是当这两项的 2 补表示的最左位数字相同，而答案的最左位数字与它们不同时。

29. 若 $m \geqslant 0$，那么首位 a_{n-1} 为 0，公式为 $m=\sum_{i=0}^{n-2} a_i 2^i$。由于右边正是 m 的二进展开，所以公式正确。若 $m<0$，它的 2 补展开以 1 为首位，其余 $n-1$ 位是 $2^{n-1}-(-m)$ 的二进展开。这表明 $2^{n-1}-(-m)=\sum_{i=0}^{n-2} a_i 2^i$。解此方程求 m 即可得想证明的等式，因为 $a_{n-1}=1$。

31. $4n$

33. **Procedure** *Cantor*（x：正整数）

```
n := 1; f := 1
while (n + 1) * f ≤ x
  begin
    n := n + 1
    f := f * n
  end
y := x
while n > 0
  begin
    a_n := ⌊y/f⌋
    y := y − a_n * f
    f := f/n
    n := n − 1
  end {x = a_n n! + a_{n−1}(n − 1)! + … + a_1 1!}
```

35. 第 1 步：$c=0$，$d=0$，$s_0=1$；第 2 步：$c=0$，$d=1$，$s_1=0$；第 3 步：$c=1$，$d=1$，$s_2=0$；第 4 步：$c=1$，$d=1$，$s_3=0$；第 5 步：$c=1$，$d=1$，$s_4=1$；第 6 步：$c=1$，$s_5=1$

37. **Procedure** *subtract*（a, b:，正整数，）$a>b$，
$a=(a_{n-1}a_{n-2}\cdots a_1a_0)_2$，$b=(b_{n-1}b_{n-2}\cdots b_1b_0)_2$）

```
B := 0 {B 是借方}
for j := 0 to n − 1
  begin
    if a_j ≥ b_j + B then
    begin
      s_j := a_j − b_j − B
      B := 0
    end
    else
    begin
```

```
            s_j := a_j + 2 - b_j - B
            B := 1
         end
      end {(s_{n-1}s_{n-2}…s_1s_0)_2 是差}
```

39. **Procedure** *compare* (a,b:正整数，且

```
      a = (a_n a_{n-1} … a_1 a_0)_2, b = (b_n b_{n-1} … b_1 b_0)_2)
      k := n
   while a_k = b_k 和 k > 0
         k := k - 1
   if a_k = b_k then 打印 "a equals b"
   if a_k > b_k then 打印 "a is greater than b"
   if a_k < b_k then 打印 "a is less than b"
```

41. $O(\log n)$

2.5 节

1. a) $1 = (-1)\cdot 10 + 1\cdot 11$
   b) $1 = 21\cdot 21 + (-10)\cdot 44$
   c) $12 = (-1)\cdot 36 + 48$
   d) $1 = 13\cdot 55 + (-21)\cdot 34$
   e) $3 = 11\cdot 213 + (-20)\cdot 117$
   f) $223 = 1\cdot 0 + 1\cdot 223$
   g) $1 = 37\cdot 2347 + (-706)\cdot 123$
   h) $2 = 1128\cdot 3454 + (-835)\cdot 4666$
   i) $1 = 2468\cdot 9999 + (-2221)\cdot 11111$
3. $15\cdot 7 = 105 \equiv 1 \pmod{26}$
5. 7
7. 52
9. 假定 b 和 c 都是 a 模 m 的逆，那么 $ba \equiv 1 \pmod m$，$ca \equiv 1 \pmod m$。因此 $ba \equiv ca \pmod m$。由于 $\gcd(a,m)=1$，由定理 2 知 $b \equiv c \pmod m$。
11. $x \equiv 8 \pmod 9$
13. 令 $m' = m/\gcd(c,m)$。由于 m 和 c 的公因子在求 m' 时都消去了，所以 m' 和 c 互素。由于 m 能整除 $ac-bc=(a-b)c$，所以 m' 整除 $(a-b)c$，由引理 1 知 m' 整除 $a-b$，所以 $a \equiv b \pmod{m'}$
15. 假定 $x^2 \equiv 1 \pmod p$，那么 p 整除 $x^2-1=(x+1)(x-1)$。由引理 2 知 $p\mid(x+1)$ 或 $p\mid(x-1)$，所以 $x \equiv -1 \pmod p$ 或 $x \equiv 1 \pmod p$。
17. a) 假定 $ia \equiv ja \pmod p$，其中 $1 \leqslant i < j < p$。于是 p 整除 $ja - ia = a(j-i)$。由定理 1 及 a 不能被 p 整除，p 能整除 $j-i$。但这不可能，因为 $j-i$ 是比 p 小的正整数。
   b) 由 a) 知 a，$2a$，…，$(p-1)a$ 中没有两个是模 p 同余的，因此各自必定与 $1 \sim p-1$

中的不同数同余。于是 $a\cdot 2a\cdot 3a\cdots(p-1)a\equiv 1\cdot 2\cdot 3\cdots(p-1)(\text{mod } p)$，即 $(p-1)!\cdot a^{p-1}\equiv(p-1)!\ (\text{mod } p)$。

c）根据 Wilson 定理和 b)，若 p 不能整除 a，必有 $(-1)\cdot a^{p-1}\equiv -1(\text{mod } p)$。因此 $a^{p-1}\equiv 1(\text{mod } p)$。

d）若 $p\mid a$，那么 $p\mid a^{p}$，因此 $a^{p}\equiv a\equiv 0\ (\text{mod } p)$。如果 p 不能整除 a，那么由 c）知 $a^{p-1}\equiv 1\ (\text{mod } p)$，两边同乘以 a，$a^{p}\equiv a\ (\text{mod } p)$。

19. 假定 p 是出现在 $m_1m_2\cdots m_n$ 的素数分解中的一个素数。由于这些 m_i 是两两互素的，p 只能是其中一个的因子，例如，是 m_j 的因子。由于 m_j 整除 $a-b$，所以 $a-b$ 的素数分解中有因子 p，而且 p 的次数至少是 m_j 的素数分解中 p 的次数。由此可知 $m_1m_2\cdots m_n$ 整除 $a-b$，所以 $a\equiv b\ (\text{mod } m_1m_2\cdots m_n)$。

21. $x\equiv 1\ (\text{mod } 6)$

23. a）由费马小定理知 $2^{10}\equiv 1\ (\text{mod } 11)$，因此 $2^{340}=(2^{10})^{34}\equiv 1^{34}=1(\text{mod } 11)$。

b）由于 $32\equiv 1\ (\text{mod } 31)$，所以 $2^{340}=(2^{5})^{68}=32^{68}\equiv 1^{68}=1(\text{mod } 31)$。

c）由于 11 和 31 互素，$11\cdot 31=341$，由 a）、b）和练习 19 知 $2^{340}\equiv 1\ (\text{mod } 341)$。

25. a）8　　b）983

27. $0=(0,\ 0)$，$1=(1,\ 1)$，$2=(2,\ 2)$，$3=(0,\ 3)$，$4=(1,\ 4)$，
$5=(2,\ 0)$，$6=(0,\ 1)$，$7=(1,\ 2)$，$8=(2,\ 3)$，$9=(0,\ 4)$，
$10=(1,\ 0)$，$11=(2,\ 1)$，$12=(0,\ 2)$，$13=(1,\ 3)$，$14=(2,\ 4)$

29. 我们有 $m_1=99$，$m_2=98$，$m_3=97$ 和 $m_4=95$，所以 $m=99\cdot 98\cdot 97\cdot 95=89403930$。进而 $M_1=m/m_1=903070$，$M_2=m/m_2=912285$，$M_3=m/m_3=921690$，$M_4=m/m_4=941094$。用欧几里德算法得出，对 $k=1,\ 2,\ 3,\ 4$ 分别有 M_k 模 m_k 的逆 $y_1=37$，$y_2=33$，$y_3=24$ 和 $y_4=4$。于是解为 $65\cdot 903\,070\cdot 37+2\cdot 912\,285\cdot 33+51\cdot 921\,690\cdot 24+10\cdot 941\,094\cdot 4=3\,397\,886\,480\equiv 537\,140\ (\text{mod } 89403930)$。

31. 由练习 30 知 $\gcd(2^{b}-1,(2^{a}-1)\ \text{mod}\ (2^{b}-1))=\gcd(2^{b}-1,2^{a\ \text{mod}\ b}-1)$。由于计算中涉及的指数是 b 和 a mod b，这与计算 $\gcd(a,b)$ 时涉及的量相同；用欧几里德算法计算 $\gcd(2^{a}-1,2^{b}-1)$ 所使用的步骤与计算 $\gcd(a,b)$ 的步骤平行执行，这就证明了 $\gcd(2^{a}-1,2^{b}-1)=2^{\gcd(a,b)}-1$。

33. 假定 q 是个奇素数且 $q\mid 2^{p}-1$。由费马小定理知 $q\mid 2^{q-1}-1$。从练习 31 知 $\gcd(2^{p}-1,2^{q-1}-1)=2^{\gcd(p,q-1)}-1$。由于 q 是 $2^{p}-1$ 和 $2^{q-1}-1$ 的公因数，$\gcd(2^{p}-1,2^{q-1}-1)>1$。于是 $\gcd(p,q-1)=p$，否则只有 $\gcd(p,q-1)=1$ 的可能，而这就导致 $\gcd(2^{p}-1,2^{q-1}-1)=1$。这样一来，$p\mid q-1$，即有正整数 m，使 $q-1=mp$。由于 q 是奇数，m 必定是偶数，如 $m=2k$，于是 $2^{p}-1$ 的每个素因数都是 $2kp+1$ 的形式。由于这种形式的若干数的乘积仍是这种形式，因而 $2^{p}-1$ 的所有除数都是这种形式的。

35. 假定我们知道 $n=pq$ 及 $(p-1)(q-1)$，要计算 p 和 q。首先注意 $(p-1)(q-1)=pq-p-q+1=n-(p+q)+1$。由此可以算出 $s=p+q$。由于 $q=s-p$，所以 $n=p(s-p)$。于是 $p^{2}-ps+n=0$。现在可以用二次方程公式求 p。一旦算出了 p，就可以算出 q，因为 $q=n/p$。

37. SILVER

39. 假定 s 是 $x^2 \equiv a \pmod p$ 的一个解。由于 $(-s)^2 = s^2$，$-s$ 也是一个解，而且 $s \not\equiv -s \pmod p$。否则 $p \mid 2s$，从而 $p \mid s$，根据 s 是解的假设，$p \mid s$ 意味着 $p \mid a$，矛盾。再假定 s 和 t 是模 p 不同余的两个解，那么由于 $s^2 \equiv t^2 \pmod p$，$p \mid (s^2 - t^2)$，即 $p \mid (s+t)(s-t)$。由引理 2，$p \mid (s-t)$ 或 $p \mid (s+t)$，所以 $s \equiv t \pmod p$，或 $s \equiv -t \pmod p$。所以至多有两个解。

41. $\left(\frac{a}{p}\right)$ 的值只依赖于 a 是不是模 p 的二次留数，即 $x^2 \equiv a \pmod p$ 是否有解。由于这只与 a 模 p 的等价类有关，所以 $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$，只要 $a \equiv b \pmod p$。

43. 由练习 42 知

$$\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = a^{(p-1)/2} b^{(p-1)/2} = (ab)^{(p-1)/2} \equiv \left(\frac{ab}{p}\right) \pmod p$$

2.6 节

1. a) 3×4 b) $\begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix}$ c) $[2 \quad 0 \quad 4 \quad 6]$ d) 1 e) $\begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 1 \\ 1 & 4 & 3 \\ 3 & 6 & 7 \end{bmatrix}$

3. a) $\begin{bmatrix} 1 & 11 \\ 2 & 18 \end{bmatrix}$ b) $\begin{bmatrix} 2 & -2 & -3 \\ 1 & 0 & 2 \\ 9 & -4 & 4 \end{bmatrix}$ c) $\begin{bmatrix} -4 & 15 & -4 & 1 \\ -3 & 10 & 2 & -3 \\ 0 & 2 & -8 & 6 \\ 1 & -8 & 18 & -13 \end{bmatrix}$

5. $\begin{bmatrix} 9/5 & -6/5 \\ -1/5 & 4/5 \end{bmatrix}$

7. $\mathbf{0} + \mathbf{A} = [0 + a_{ij}] = [a_{ij} + 0] = \mathbf{0} + \mathbf{A}$

9. $\mathbf{A} + (\mathbf{B} + \mathbf{C}) = [a_{ij} + (b_{ij} + c_{ij})]$
$= [(a_{ij} + b_{ij}) + c_{ij}]$
$= (\mathbf{A} + \mathbf{B}) + \mathbf{C}$

11. **A** 的行数等于 **B** 的列数，且 **A** 的列数等于 **B** 的行数。

13. $\mathbf{A}(\mathbf{BC}) = [\sum_q a_{iq}(\sum_r b_{qr} c_{rl})]$
$= [\sum_q \sum_r a_{iq} b_{qr} c_{rl}]$
$= [\sum_r \sum_q a_{iq} b_{qr} c_{rl}]$
$= [\sum_r (\sum_q a_{iq} b_{qr}) c_{rl}]$
$= (\mathbf{AB})\mathbf{C}$

15. $\mathbf{A}^n = \begin{bmatrix} 1 & n \\ 0 & 1 \end{bmatrix}$

17. a) 令 $\mathbf{A} = [a_{ij}]$，$\mathbf{B} = [b_{ij}]$，那么 $(\mathbf{A} + \mathbf{B}) = [a_{ij} + b_{ij}]$。有 $(\mathbf{A} + \mathbf{B})^t = [a_{ji} + b_{ji}] = [a_{ji}] + [b_{ji}] = \mathbf{A}^t + \mathbf{B}^t$。

b) 用 a) 中同样的记号，有 $\mathbf{B}^t\mathbf{A}^t = \left[\sum_q b_{qi}a_{jq}\right] = \left[\sum_q a_{jq}b_{qi}\right] = (\mathbf{AB})^t$，因为这个矩阵的 (i,j) 位元素是 $\mathbf{AB}$ 的 (j,i) 位元素。

19. 由于 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \begin{pmatrix} ab-bc & 0 \\ 0 & ab-bc \end{pmatrix}$

$$= (ad-bc)\mathbf{I}_2$$

$$= \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

结论成立

21. 由结合律知 $\mathbf{A}^n(\mathbf{A}^{-1})^n = \mathbf{A}(\mathbf{A}\cdots(\mathbf{A}(\mathbf{AA}^{-1})\mathbf{A}^{-1})\cdots\mathbf{A}^{-1})^{-1}$。由于 $\mathbf{AA}^{-1} = I$，从内向外结合可知 $\mathbf{A}^n(\mathbf{A}^{-1})^n = I$。类似地，$(\mathbf{A}^{-1})^n\mathbf{A}^n = I$。因此 $(\mathbf{A}^n)^{-1} = (\mathbf{A}^{-1})^n$。

23. 需用 m_2 次乘法来计算乘积中 m_1m_3 个元素中的每一个，所以共有 $m_1m_2m_3$ 次乘法。

25. $\mathbf{A}_1((\mathbf{A}_2\mathbf{A}_3)\mathbf{A}_4)$

27. $x_1 = 1, x_2 = -1, x_3 = -2$

29. a) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

31. a) $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

33. a) $\mathbf{A} \vee \mathbf{B} = [a_{ij} \vee b_{ij}] = [b_{ij} \vee a_{ij}] = \mathbf{B} \vee \mathbf{A}$

b) $\mathbf{A} \wedge \mathbf{B} = [a_{ij} \wedge b_{ij}] = [b_{ij} \wedge a_{ij}] = \mathbf{B} \wedge \mathbf{A}$

35. a) $\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C}) = [a_{ij}] \vee [b_{ij} \wedge c_{ij}]$

$$= [a_{ij} \vee (b_{ij} \wedge c_{ij})]$$

$$= [(a_{ij} \vee b_{ij}) \wedge (a_{ij} \vee c_{ij})]$$

$$= [a_{ij} \vee b_{ij}] \wedge [a_{ij} \vee c_{ij}]$$

$$= (\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C})$$

b) $\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}) = [a_{ij}] \wedge [b_{ij} \vee c_{ij}]$

$$= [a_{ij} \wedge (b_{ij} \vee c_{ij})]$$

$$= [(a_{ij} \wedge b_{ij}) \vee (a_{ij} \wedge c_{ij})]$$

$$= [a_{ij} \wedge b_{ij}] \vee [a_{ij} \wedge c_{ij}]$$

$$= (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})$$

37. $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = \left[\bigvee_q a_{iq} \wedge \left(\bigvee_r (b_{qr} \wedge c_{rl})\right)\right]$

$$= \left[\bigvee_q \bigvee_r (a_{iq} \wedge b_{qr} \wedge c_{rl})\right]$$

$$= \left[\bigvee_r \bigvee_q (a_{iq} \wedge b_{qr} \wedge c_{rl})\right]$$

$$= \left[\bigvee_r \left(\bigvee_q (a_{iq} \wedge b_{qr})\right) \wedge c_{rl}\right]$$

$$= (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$$

补充练习

1. a)
```
Procedure last max (a_1,..., a_n: 整数)
max := a_1
last := 1
i := 2
while i ≤ n
  begin
    if a_i ≥ max then
      begin
        max := a_i
        last := i
      end
    i := i + 1
  end last 是列表中最大整数最后出现的位置
```
b) $2n-1=O(n)$ 比较。

3. a)
```
Procedure pair zeros (b_1 b_2 ··· b_n: 位串, n ≥ 2)
 x := b_1
 y := b_2
 k := 2
 while (k < n 和(x ≠ 0 或 y ≠ 0))
 begin
   k := k + 1
   x := y
   y := b_k
 end
 if(x = 0 和 y = 0) then 打印"YES"
else 打印 "NO"
```
b) $O(n)$ 比较。

5. 5，22，-12，-29

7. 由于 $ac\equiv bc\pmod m$，必有整数 k 使 $ac=bc+km$。因此 $a-b=km/c$。由于 $a-b$ 为整数，$c\mid km$。令 $d=\gcd(m,c)$，于是 $c=de$。由于 e 没有能整除 m/d 的因子，所以 $d\mid m$，$e\mid k$。这样 $a-b=(k/e)(m/d)$，其中 $k/e\in\mathbf{Z}$，$m/d\in\mathbf{Z}$。于是 $a\equiv b\pmod{m/d}$ 。

9. 1

11. 1

13. $(a_n a_{n-1}\cdots a_1 a_0)_{10}=\sum_{k=0}^{n}10^k\cdot a_k\equiv\sum_{k=0}^{n}a_k\pmod 9$，这是因为 $10^k\equiv 1\pmod 9$ 对每个非负整数 k 均成立。

15. a) 不互素。
    b) 互素。
    c) 互素。
    d) 互素。
17. a) 解密函数是 $g(q)=\overline{a}(q-b)\bmod 26$，其中 $\overline{a}$ 是 a 模 26 的一个逆。
    b) PLEASE SEND MONEY
19. $x \equiv 28 \pmod{30}$
21. 对 $n \geqslant 0$

$$\mathbf{A}^{4n}=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix},\quad \mathbf{A}^{4n+1}=\begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix},\quad \mathbf{A}^{4n+2}=\begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix},\quad \mathbf{A}^{4n+3}=\begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}$$

23. 假定

$$\mathbf{A}=\begin{bmatrix}a & b\\c & d\end{bmatrix}。$$

令

$$\mathbf{B}=\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}$$

由于 $\mathbf{AB}=\mathbf{BA}$，所以 $c=0$ 且 $a=d$。令

$$\mathbf{B}=\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}$$

由于 $\mathbf{AB}=\mathbf{BA}$，所以 $b=0$。因此

$$\mathbf{A}=\begin{bmatrix}a & 0\\0 & a\end{bmatrix}=a\mathbf{I}$$

25. **Procedure** *triangular matrix multiplication* ($\mathbf{A}$，$\mathbf{B}$：上三角形 $n\times n$ 矩阵，$\mathbf{A}=[a_{ij}]$，$\mathbf{B}=[b_{ij}]$)

```
for i := 1 to n
  begin
    for j := i to n
      begin
        c_ij := 0
        for k := i to j
          c_ij := c_ij + a_ik b_kj
      end
  end
```

27. $(\mathbf{AB})(\mathbf{B}^{-1}\mathbf{A}^{-1})=\mathbf{A}(\mathbf{BB}^{-1})\mathbf{A}^{-1}=\mathbf{AIA}^{-1}=\mathbf{AA}^{-1}=\mathbf{I}$。类似地，$(\mathbf{B}^{-1}\mathbf{A}^{-1})(\mathbf{AB})=\mathbf{I}$。所以 $(\mathbf{AB})^{-1}=\mathbf{B}^{-1}\mathbf{A}^{-1}$。
29. a) 令 $\mathbf{A}\odot\mathbf{0}=[b_{ij}]$，那么 $b_{ij}=(a_{i1}\wedge 0)\vee\cdots\vee(a_{ip}\wedge 0)=0$，因此 $\mathbf{A}\odot\mathbf{0}=\mathbf{0}$。类似地，$\mathbf{0}\odot\mathbf{A}=\mathbf{0}$。
    b) $\mathbf{A}\vee\mathbf{0}=[a_{ij}\vee 0]=[a_{ij}]=A$，因此 $\mathbf{A}\vee\mathbf{0}=\mathbf{A}$。类似地，$\mathbf{0}\vee\mathbf{A}=\mathbf{A}$。
    c) $\mathbf{A}\wedge\mathbf{0}=[a_{ij}\wedge 0]=[0]=0$，因此 $\mathbf{A}\wedge\mathbf{0}=\mathbf{0}$。类似地，$\mathbf{0}\wedge\mathbf{A}=\mathbf{0}$。

第 3 章

3.1 节

1. a) 附加 b) 化简 c) 假言推理 d) 拒取式 e) 假言三段论
3. 设 w 是"兰迪努力工作"，设 d 是"兰迪是个笨小子"，设 j 是"兰迪将得到这份工作"。前提是 w，$w\to d$ 和 $d\to\neg j$。用假言推理和前两个前提得出 d。用假言推理和最后一个前提得出 $\neg j$，它就是所需要的结论："兰迪将得不到这份工作"。
5. 用全称量词消去来得出"若苏格拉底是人，则苏格拉底是要死的"。用假言推理来得出苏格拉底是要死的。
7. a) 有效结论是"我周二没有休假"，"我周四休过假"，"周四下过雨"。
   b) "我没有吃辣的食物而且没有打雷"是有效结论。
   c) "我是聪明的"是有效结论。
   d) "拉尔夫不是主修计算机科学的"是有效结论。
   e) "你购买许多东西对美国有利而且对你有利"是有效结论。
   f) "老鼠啃咬它们的食物"和"兔子不是鼠类"是有效结论。
9. a) 设 $c(x)$ 是"x 是在这个班里"，$j(x)$ 是"x 知道如何用 JAVA 编写程序"，$h(x)$ 是"x 可以找到高薪工作"。前提是 c(道格)，j(道格)，$\forall x(j(x)\to h(x))$。用全称量词消去和最后一个前提得出 j(道格)$\to h$(道格)。对这个结论和第 2 个前提用假言推理得出 h(道格)。用合取和第 1 个前提得出 c(道格)$\wedge h$(道格)。最后，用存在量词引入得出所需要的结论 $\exists x(c(x)\wedge h(x))$。
   b) 设 $c(x)$ 是"x 是在这个班里"，$w(x)$ 是"x 喜欢鲸鱼观察"，$p(x)$ 是"x 关心海洋污染"。前提是 $\exists x(c(x)\wedge w(x))$ 和 $\forall x(w(x)\to p(x))$。根据第 1 个前提，对一个具体的人 y 来说有 $c(y)\wedge w(y)$。化简得出 $w(y)$。用第 2 个前提和全称量词消去得出 $w(y)\to p(y)$。用假言推理得出 $p(y)$，再用合取得出 $c(y)\wedge p(y)$。最后，用存在量词引入得出所需要的结论 $\exists x(c(x)\wedge p(x))$。
   c) 设 $c(x)$ 是"x 是在这个班里"，$p(x)$ 是"x 拥有一台个人电脑"，$w(x)$ 是"x 会使用字处理程序"。前提是 c(齐克)，$\forall x(c(x)\to p(x))$ 和 $\forall x(p(x)\to w(x))$。用第 2 个前提和全称量词消去得出 c(齐克)$\to p$(齐克)。用第 1 个前提和假言推理得出 p(齐克)。用第 3 个前提和全称量词消去得出 p(齐克)$\to w$(齐克)。最后，用假言推理得出所需要的结论 w(齐克)。
   d) 设 $j(x)$ 是"x 是在新泽西"，$f(x)$ 是"x 住在距离大海 50 英里之内"，$s(x)$ 是"x 见过大海"。前提是 $\forall x(j(x)\to f(x))$ 和 $\exists x(j(x)\wedge\neg s(x))$。第 2 个前提和存在量词消去意味着对一个具体的人 y 来说有 $j(y)\wedge\neg s(y)$。化简得出对这个人 y 来说有 $j(y)$。用全称量词消去和第 1 个前提得出 $j(y)\to f(y)$，再用假言推理得出 $f(y)$。用化简从 $j(y)\wedge\neg s(y)$ 得出 $\neg s(y)$。所以用合取得出 $f(y)\wedge\neg s(y)$。最后，用存在量词引入得出所需要的结论 $\exists x(f(x)\wedge\neg s(x))$。
11. a) 肯定结论谬误
    b) 回避问题谬误

c) 使用假言推理的有效论证

d) 使用析取三段论的有效论证

e) 否定前提谬误

13. 这个命题平凡地为真，因为 0 不是正整数。空证明。

15. $P(1)$为真，因为 $(a+b)^1=a+b\geqslant a^1+b^1=a+b$。直接证明。

17. a) 假定 n 是奇数，所以对某个整数 k 来说有 $n=2k+1$。于是 $n^3+5=2(4k^3+6k^2+3k+3)$。因为 n^3+5 是 2 乘以某个整数，所以它是偶数。

b) 假设 n^3+5 是奇数而且 n 是奇数。因为 n 是奇数并且两个奇数之积是奇数，所以 n^2 是奇数而且 n^3 也是奇数。但是这样一来$5=(n^3+5)-n^3$ 应当是偶数，因为它是两个奇数之差。因此，n^3+5 和 n 都是奇数这个假设是错误的。

19. 设 $n=2k+1$ 和 $m=2l+1$ 都是奇数，则 $n+m=2(k+l+1)$ 是偶数。

21. 假设 r 是有理数，i 是无理数，$s=r+i$ 是有理数。则根据练习 10，$s+(-r)=i$ 是有理数，矛盾。

23. 因为$\sqrt{2}\cdot\sqrt{2}=2$ 是有理数，且$\sqrt{2}$是无理数，所以两个无理数之积不一定是无理数。

25. $n=1601$ 是一个反例。

27. 假设 $3^{1/3}=a/b$，其中$a,b\in Z$，$b\neq0$，而且$\gcd(a,b)=1$。于是 $3=a^3/b^3$，所以 $3b^3=a^3$。因此 $3\mid a^3$，且仅当 $3\mid a$ 时才可以发生这样的事。设 $a=3m$，则 $3b^3=27m^3$，或 $b^3=9m^3$。因此 $3\mid b^3$，这说明 $3\mid b$。这与假设的$\gcd(a,b)=1$ 矛盾。

29. 若 $x\leqslant y$，则$\max(x,y)+\min(x,y)=y+x=x+y$。若 $x\geqslant y$，则 $\max(x,y)+\min(x,y)=x+y$。因为这是仅有的两种情况，所以不等式总是成立。

31. 存在四种情形。情形 1：$x\geqslant0$ 而且 $y\geqslant0$，于是$|x|+|y|=x+y=|x+y|$。情形 2：$x<0$ 而且 $y<0$，于是$|x|+|y|=-x+(-y)=-(x+y)=|x+y|$，因为 $x+y<0$。情形 3：$x\geqslant0$ 而且 $y<0$，于是$|x|+|y|=x+(-y)$。若 $x\geqslant-y$，则$|x+y|=x+y$。但是因为 $y<0$，$-y>y$，所以$|x|+|y|=x+(-y)>x+y=|x+y|$。若 $x<-y$，则$|x+y|=-(x+y)=-x+(-y)$。但是因为 $x\geqslant0$，$-x\leqslant x$，所以$|x|+|y|=x+(-y)\geqslant-x+(-y)=|x+y|$。情形 4：$x<0$ 而且 $y\geqslant0$。交换 x 与 y，就等同于情形 3。

33. 有三种情形需要考虑：情形 1，a 是最小的，或是平局最小的；情形 2，b 是最小的，或是平局最小的；情形 3，c 是最小的，或是平局最小的。因为 a，b 和 c 中有一个是最小的，或是平局最小的，所以这三种情形覆盖了所有的可能性。在情形 1 里，$a\leqslant\min(b,c)$，所以左边是 a 而且右边也是a，因为$\min(a,c)=a$。其余两种情形里的论证是类似的。

35. 首先，假定 n 是奇数，所以对某个整数 k 来说$n=2k+1$。于是 $5n+6=5(2k+1)+6=10k+11=2(5k+5)+1$。因此 $5n+6$ 是奇数。为了证明逆命题，假定 n 是偶数，所以对某个整数 k 来说有 $n=2k$。于是 $5n+6=10k+6=2(5k+3)$，所以 $5n+6$ 是偶数。因此 n 是奇数当且仅当 $5n+6$ 是奇数。

37. $a^2\equiv b^2\pmod p$当且仅当$p\mid(a^2-b^2)=(a+b)(a-b)$。根据素数分解的唯一性，这等价于$p\mid(a-b)$或 $p\mid(a+b)$，这与 $a\equiv b\pmod p$或 $a\equiv-b\pmod p$是一样的。

39. 这个命题为真。假定 m 既不是 1 也不是 -1。则 mn 有大于 1 的因子 m。在另一方面，

$mn=1$，且 1 没有这样的因子。因此 $m=1$ 或 $m=-1$。在第 1 种情形里 $n=1$，而在第 2 种情形里 $n=-1$，因为 $n=1/m$。

41. 正整数 3 不是两个整数的平方和，所以这个命题为假。

43. a) 真；因为$\lfloor x \rfloor$已经是整数，所以$\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$。
    b) 假；$x=1/2$ 是反例。
    c) 真；若 x 或 y 是整数，则用 1.6 节表 1-1 里的性质 4b，差为 0。若 x 与 y 都不是整数，则 $x=n+\epsilon$，$y=m+\delta$，其中 n 和 m 都是整数且 ϵ 和 δ 都是小于 1 的正实数。于是 $m+n<x+y<m+n+2$，所以 $\lceil x+y \rceil$或者是 $m+n+1$ 或者是 $m+n+2$。因此，所给的表达式或者是$(n+1)+(m+1)-(m+n+1)=1$ 或者是$(n+1)+(m+1)-(m+n+2)=0$，正如所需要的那样。
    d) 假；$x=\frac{1}{4}$和 $y=3$ 是反例。
    e) 假；$x=\frac{1}{2}$是反例。

45. a) 若 x 是正整数，则两边是相等的。所以假定 $x=n^2+m+\epsilon$，其中 n^2 是小于 x 的最大完全平方数，m 是非负整数，而 $0<\epsilon\leqslant 1$。则$\sqrt{x}$和$\sqrt{\lfloor x \rfloor}=\sqrt{n^2+m}$都是在 n 和 $n+1$ 之间，所以两边都等于 n。
    b) 若 x 是正整数，则两边是相等的。所以假定 $x=n^2-m-\epsilon$，其 n^2 是大于 x 的最小完全平方数，m 是非负整数，而 ϵ 是满足 $0<\epsilon\leqslant 1$ 的实数。则$\sqrt{x}$和$\sqrt{\lceil x \rceil}=\sqrt{n^2-m}$都是在 $n-1$ 和 n，所以两边都等于 n。

47. 设 $x=n+(r/m)+\epsilon$，其中 n 是整数，r 是小于 m 的非负整数，m 是非负整数，而 ϵ 是满足 $0\leqslant\epsilon<1/m$ 的实数。左边是$\lfloor nm+r+m\epsilon \rfloor=nm+r$。在右边，从$\lfloor x \rfloor$到$\lfloor x+(m+r-1)/m \rfloor$的项都恰好是 n，而从$\lfloor x+(m-r)/m \rfloor$开始的项都是 $n+1$。因此，右边是$(m-r)n+r(n+1)=nm+r$，与左边一样。

49. 将给出归谬证明。假定 a_1，a_2，…，a_n 都小于 A，其中 A 是这些数的平均值。则 $a_1+a_2+\cdots+a_n<nA$。两边都除以 n 就证明$A=(a_1+a_2+\cdots+a_n)/n<A$，矛盾。

51. 将通过证明(i)蕴涵(ii)、(ii)蕴涵(iii)、(iii)蕴涵(iv)和(iv)蕴涵（i）来证明这四个命题都是等价的。首先，假设 n 是偶数。则对某个整数 k 来说$n=2k$。于是 $n+1=2k+1$，所以 $n+1$ 是奇数。这证明(i)蕴涵(ii)。其次，假设 $n+1$ 是奇数。则对某个整数 k 来说 $n+1=2k+1$。于是$3n+1=2n+(n+1)=2(n+k)+1$，这说明 $3n+1$ 是奇数，证明(ii)蕴涵(iii)。再次，假设 $3n+1$ 是奇数。则对某个整数 k 来说 $3n+1=2k+1$。于是 $3n=(2k+1)-1=2k$，所以 $3n$ 是偶数。这证明(iii)蕴涵(iv)。最后，假设 n 不是偶数。则 n 是奇数，所以对某个整数 k 来说 $n=2k+1$。于是$3n=3(2k+1)=6k+3=2(3k+1)+1$，所以 $3n$ 是奇数。这样完成了(iv)蕴涵(i)的间接证明。

53. 整数 3，5 和 7 是三个属于所需要的形式的素数。

55. 根据第二个前提，存在某个不喝咖啡的狮子。设里欧是这样的动物。通过化简知道里欧是狮子。通过假言推理，从第 1 个前提知道里欧是凶猛。因此里欧是凶猛的并且是不喝咖啡的。根据存在量词的定义，存在着不喝咖啡的凶猛的动物，即某些不喝咖啡的凶猛动物。

57. 假设有前 $n+1$ 个素数 p_1，p_2，…，p_{n+1}，则 $p_1p_2\cdots p_{n+1}$ 被超过 n 个素数整除。
59. 假设 p_1，p_2，$\cdots p_n$ 是所有模 4 同余 3 的素数，3 除外。设 $q=4p_1p_2\cdots p_n+3$。则 $q\equiv 3 \pmod 4$，并且 q 不能被 p_i 或 3 整除，$i=1$，$2\cdots$，n。因为 q 至少有一个模 4 同余 3 的素因子，所以必然存在一个这种类型的素数，它不在这个列表里。这是一个非构造性的存在性证明。
61. 假设 $p_1\to p_4\to p_2\to p_5\to p_3\to p_1$。为了证明这些命题中的一个蕴涵着其余的任意一个，只需要重复使用假言三段论。
63. 设 $a=\sqrt{2}$ 和 $b=\sqrt{2}$。若 $c=a^b$ 是有理数，则大功告成。若 c 是无理数，则 $c^b=(a^b)^b=(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}}=(\sqrt{2})^2=2$ 是有理数。这个证明是非构造性的。
65. 每个放在棋盘上的多米诺骨牌都恰好覆盖一个白格和一个黑格，因此一组多米诺骨牌恰好覆盖同样数目的白格和黑格。因为删除对角上的格子产生出黑格比白格多两个或白格比黑格多两个的棋盘，所以没有一组多米诺骨牌可以覆盖删除了对角上的格子的棋盘。
67. 有效。

3.2 节

1. $n(n+1)$
3. 设 $P(n)$ 是 “$\sum_{j=0}^{n}3\cdot 5^j=3(5^{n+1}-1)/4$”。基础步骤：$P(0)$ 为真，因为 $\sum_{j=0}^{0}3\cdot 5^j=3=3(5^1-1)/4$。归纳步骤：假设 $\sum_{j=0}^{n}3\cdot 5^j=3(5^{n+1}-1)/4$。则 $\sum_{j=0}^{n+1}3\cdot 5^j=(\sum_{j=0}^{n}3\cdot 5^j)+3\cdot 5^{n+1}=3(5^{n+1}-1)/4+3\cdot 5^{n+1}=3(5^{n+1}+4\cdot 5^{n+1}-1)/4=3(5^{n+2}-1)/4$。
5. 通过检查小的 n 值做出猜想：$P(n)$ 为真，其中 $P(n)$ 是命题 “$\sum_{j=1}^{n}1/2^j=(2^n-1)/2^n$”。基础步骤：$P(1)$ 为真，因为 $1/2=(2^1-1)/2^1$。归纳步骤：假设 $\sum_{j=1}^{n}1/2^j=(2^n-1)/2^n$，则

$$\sum_{j=1}^{n+1}\frac{1}{2^j}=\left(\sum_{j=1}^{n}\frac{1}{2^j}\right)+\frac{1}{2^{n+1}}=\frac{2^n-1}{2^n}+\frac{1}{2^{n+1}}=\frac{2^{n+1}-2+1}{2^{n+1}}=\frac{2^{n+1}-1}{2^{n+1}}$$

7. 设 $P(n)$ 是 “$\sum_{j=1}^{n}j^2=n(n+1)(2n+1)/6$”。基础步骤：$P(1)$ 为真，因为 $\sum_{1}^{1}j^2=1=1(1+1)(2\cdot 1+1)/6$。归纳步骤：假设 $\sum_{j=1}^{n}j^2=n(n+1)(2n+1)/6$。则 $\sum_{j=1}^{n+1}j^2=(\sum_{j=1}^{n}j^2)+(n+1)^2=n(n+1)(2n+1)/6+(n+1)^2=(n+1)[2n^2+n+6n+6]/6=(n+1)(n+2)\cdot(2n+3)/6=(n+1)((n+1)+1)(2(n+1)+1)/6$。
9. 设 $P(n)$ 是 “$1^2+3^2+\cdots+(2n+1)^2=(n+1)(2n+1)(2n+3)/3$”。基础步骤：$P(0)$ 为真，因为 $1^2=1=(0+1)(2\cdot 0+1)(2\cdot 0+3)/3$。归纳步骤：假设 $P(n)$ 为真。则 $1^2+3^2+\cdots+(2n+1)^2+(2(n+1)+1)^2=(n+1)(2n+1)(2n+3)/3+(2n+3)^2=(2n+3)[(n+1)(2n+1)/3+(2n+3)]=(2n+3)(2n^2+9n+10)/3=(2n+3)(2n+5)(n+2)/3=((n+1)+1)(2(n+1)+1)(2(n+1)+3)/3$。
11. 设 $P(n)$ 是“$1+nh\leqslant(1+h)^n, h>-1$”。基础步骤：$P(0)$ 为真，因为 $1+0\cdot h=1\leqslant 1=(1+h)^0$。归纳步骤：假设 $1+nh\leqslant(1+h)^n$。则因为 $(1+h)>0$，所以 $(1+h)^{n+1}=(1+h)(1+h)^n\geqslant(1+h)(1+nh)=1+(n+1)h+nh^2\geqslant 1+(n+1)h$。

13. 设$P(n)$是"$2^n > n^2$"。基础步骤：$P(5)$为真，因为$2^5=32>25=5^2$。归纳步骤：假设$P(n)$为真，即$2^n>n^2$。则$2^{n+1}=2\cdot 2^n>n^2+n^2>n^2+4n\geq n^2+2n+1=(n+1)^2$，因为$n>4$。

15. 设$P(n)$是"$1\cdot 2+2\cdot 3+\cdots+n(n+1)=n(n+1)(n+2)/3$"。基础步骤：$P(1)$为真，因为$1\cdot 2=2=1(1+1)(1+2)/3$。归纳步骤：假设$P(n)$为真。则$1\cdot 2+2\cdot 3+\cdots+n(n+1)+(n+1)(n+2)=[n(n+1)(n+2)/3]+(n+1)(n+2)=(n+1)(n+2)[(n/3)+1]=(n+1)(n+2)(n+3)/3$。

17. 设$P(n)$是"$1^2-2^2+3^2-\cdots+(-1)^{n-1}n^2=(-1)^{n-1}n(n+1)/2$"。基础步骤：$P(1)$为真，因为$1^2=1=(-1)^0 1^2$。归纳步骤：假设$P(n)$为真。则$1^2-2^2+3^2-\cdots+(-1)^{n-1}n^2+(-1)^n(n+1)^2=(-1)^{n-1}n(n+1)/2+(-1)^n(n+1)^2=(-1)^n(n+1)[-n/2+(n+1)]=(-1)^n(n+1)[(n/2)+1]=(-1)^n(n+1)(n+2)/2$。

19. 设$P(n)$是"n分邮资可以用3分和5分邮票来组成"。基础步骤：$P(8)$为真，因为8分邮资可以用一个3分和一个5分邮票来组成。归纳步骤：假设$P(n)$为真，即可以组成n分邮资。下面将说明如何组成$n+1$分邮资。根据归纳假设可以组成n分邮资。若这里面包含一个5分邮票，则用两个3分邮票替换这个5分邮票来得到$n+1$分邮资。否则，里面只有3分邮票并且$n\geq 9$。删除三个3分邮票，用两个5分邮票替换它们来得到$n+1$分邮资。

21. 设$P(n)$是"n^5-n能被5整除"。基础步骤：$P(0)$为真，因为$0^5-0=0$能被5整除。归纳步骤：假设$P(n)$为真，即n^5-n能被5整除，则$(n+1)^5-(n+1)=(n^5+5n^4+10n^3+10n^2+5n+1)-(n+1)=(n^5-n)+5(n^4+2n^3+2n^2+n)$也能被5整除，因为在这个和里的两项都能被5整除。

23. 设$P(n)$是命题：$(2n-1)^2-1$能被8整除。基础情形$P(1)$为真，因为$8|0$。现在假设$P(n)$为真。因为$((2(n+1)-1)^2+1)=((2n-1)^2-1)+8n$，所以$P(n+1)$为真，因为右边两项都能被8整除。这说明对所有正整数来说$P(n)$都为真，所以每当m是奇正整数时，就有m^2-1能被8整除。

25. 设$P(n)$是命题：带n个元素的集合有$n(n-1)/2$个双元素子集合。基础情形$P(2)$为真，因为带2个元素的集合有一个双元素子集合，即它自身，且$2(2-1)/2=1$。现在假设$P(n)$为真。设S是带$n+1$个元素的集合。选择S里的一个元素a并且设$T=S-\{a\}$。S的双元素子集合要么包含a要么不包含a。不包含a的那些子集合都是T的2元素子集合，根据归纳假设，有$n(n-1)/2$个这样的子集合。包含a的S的双元素子集合有n个，因为这样的子集合包含a和T里的n个元素之一。因此有$n(n-1)/2+n=n(n+1)/2$个S的双元素子集合。这样就完成了归纳证明。

27. 设$P(n)$是命题：$1^4+2^4+3^4+\cdots+n^4=n(n+1)(2n+1)(3n^2+3n-1)/30$。$P(1)$为真，因为$1\cdot 2\cdot 3\cdot 5/30=1$。假设$P(n)$为真。则$(1^4+2^4+3^4+\cdots+n^4)+(n+1)^4=n(n+1)(2n+1)(3n^2+3n-1)/30+(n+1)^4=((n+1)/30)(n(2n+1)(3n^2+3n-1)+30(n+1)^3)=((n+1)/30)(6n^4+39n^3+91n^2+89n+30)=((n+1)/30)(n+2)(2n+3)(3(n+1)^2+3(n+1)-1)$。这说明$P(n+1)$为真。

29. 通过检查发现，对$n=0, 1, 2, 3$来说，不等式$2n+3\leq 2^n$不成立。设$P(n)$是命题：对正整数n来说，这个不等式成立。基础情形$P(4)$为真，因为$2\cdot 4+3=11\leq 16=$

2^4。归纳步骤：假设 $P(n)$ 为真。则根据归纳假设，$2(n+1)+3=(2n+3)+2<2^n+2$。但是因为 $n\geqslant 1$，所以 $2^n+2\leqslant 2^n+2^n=2^{n+1}$。这说明$P(n+1)$为真。

31. a) 用 5 分和 6 分邮票可以组成的邮资是 5 分、6 分、10 分、11 分、12 分、15 分、16 分、17 分、18 分以及 20 分以上的所有邮资。

b) 证明 20 分以上的所有邮资都可以用 5 分和 6 分邮票来组成。设$P(n)$是命题：可以组成 n 分邮资。$P(20)$为真，因为 20 分邮资可以用四个 5 分邮票来组成。现在假设$P(n)$为真。若使用了一个 5 分邮票来组成 n 分邮资，则用一个 6 分邮票替换它来组成 $n+1$ 分邮资。否则，若只用过 6 分邮票，则因为 $n\geqslant 20$，所以至少用过四个 6 分邮票。用五个 5 分邮票替换四个 6 分邮票来得到 $n+1$ 分邮资。因为$P(n+1)$为真。这样就完成了数学归纳法证明。

c) 设$P(n)$与在 b) 里一样。基础情形是$P(20)$, $P(21)$, $P(22)$, $P(23)$和 $P(24)$。这些情形都为真，因为 20 分、21 分、22 分、23 分和 24 分邮资分别可以用四个 5 分邮票、三个 5 分邮票和一个 6 分邮票、两个 5 分邮票和两个 6 分邮票、一个 5 分邮票和三个 6 分邮票以及四个 6 分邮票来组成。现在假设对 $20\leqslant k\leqslant n$ 来说$P(k)$为真，其中 $n\geqslant 24$。因为 $n+1\geqslant 25$，所以 $n-4\geqslant 20$，所以根据归纳假设，可以组成 $n-4$ 分邮资。添加一个 5 分邮票来得到 $n+1$ 分邮资，这说明$P(n+1)$为真。这样就完成了用数学归纳法第二原理的证明。

33. 所有大于或等于$ 40 的$ 10 的倍数，以及$ 20，都可以组成。设$P(n)$是命题：可以组成 $10n$ 美元。$P(4)$为真，因为用两个$ 20 可以组成$ 40。现在假设对 $n\geqslant 4$ 来说$P(n)$为真。若用了一个$ 50 钞票来组成 $10n$ 美元，则用三个$ 20 钞票替换它来得到$10(n+1)$美元。否则，至少用过两个$ 20 钞票，因为 $10n$ 至少为$ 40。用一个$ 50 钞票替换两个$ 20 钞票来得到$ $10(n+1)$。这说明$P(n+1)$为真。

35. 对 $n=1$ 来说，左边只是 1/1，它是 1。对 $n=2$ 来说，存在三个非空子集合 {1}，{2} 和 {1，2}，所以左边是$1/1+1/2+1/(1\cdot 2)=2$。为了证明归纳步骤，假设对 n 来说命题为真。前 $n+1$ 个正整数的集合有许多非空子集合，但它们分成三个类别：前 n 个正整数的非空子集合，前 n 个正整数的非空子集合加上 $n+1$，或仅仅 $\{n+1\}$。根据归纳假设，第 1 个类别之和为 n。对第 2 个类别来说，可以从和的每项里分解出$1/(n+1)$来，根据归纳假设，剩下的和恰是 n，所以这个部分的和是$n/(n+1)$。最后，第 3 个类别只产生出$1/(n+1)$。因此，总和为$n+n/(n+1)+1/(n+1)=n+1$。

37. 基础情形 $n=0$ 和 $n=1$ 都为真，因为 x^0 的导数是 0 且 $x^1=x$ 的导数是 1。利用乘法规则，归纳假设以及基础情形，就可证明$\frac{d}{dx}x^{n+1}=\frac{d}{dx}(x\cdot x^n)=x\cdot\frac{d}{dx}x^n+x^n\frac{d}{dx}x=x\cdot nx^{n-1}+x^n\cdot 1=nx^n+x^n=(n+1)x^n$。

39. 设$P(n)$是命题：$\mathbf{AB}^n=\mathbf{B}^n\mathbf{A}$。$P(1)$为真，因为 $\mathbf{AB}=\mathbf{BA}$。现在假设$P(n)$为真。则 $\mathbf{AB}^{n+1}=\mathbf{AB}^n\mathbf{B}=\mathbf{B}^n\mathbf{AB}=\mathbf{B}^n\mathbf{BA}=\mathbf{B}^{n+1}\mathbf{A}$。所以$P(n+1)$为真。

41. 设$P(n)$是"$(A_1\cup A_2\cup\cdots\cup A_n)\cap B=(A_1\cap B)\cup(A_2\cap B)\cup\cdots\cup(A_n\cap B)$"。基础步骤：$P(1)$平凡地为真。归纳步骤：假设 $P(n)$ 为真，则$(A_1\cup A_2\cup\cdots\cup A_{n+1})\cap B=[(A_1\cup A_2\cup\cdots\cup A_n)\cup A_{n+1}]\cap B=[(A_1\cup A_2\cup\cdots\cup A_n)\cap B]\cup(A_{n+1}\cap B)=[(A_1\cap B)\cup(A_2\cap B)\cup\cdots\cup(A_n\cap B)]\cup(A_{n+1}\cap B)=(A_1\cap B)\cup(A_2\cap B)\cup\cdots\cup(A_n\cap B)\cup$

$(A_{n+1} \cap B)$。

43. 设$P(n)$是

"$\overline{\bigcup_{k=1}^{n} A_k} = \bigcap_{k=1}^{n} \overline{A_k}$"

基础步骤：$P(1)$平凡地为真。归纳步骤：假设$P(n)$为真，则

$$\overline{\bigcup_{k=1}^{n+1} A_k} = \overline{(\bigcup_{k=1}^{n} A_k) \cup A_{n+1}} = \overline{(\bigcup_{k=1}^{n} A_k)} \cap \overline{A_{n+1}} = (\bigcap_{k=1}^{n} \overline{A_k}) \cap \overline{A_{n+1}} = \bigcap_{k=1}^{n+1} \overline{A_k}$$

45. 设$P(n)$是"$[(p_1 \to p_2) \land (p_2 \to p_3) \land \cdots \land (p_{n-1} \to p_n)] \to [(p_1 \land p_2 \land \cdots \land p_{n-1}) \to p_n]$"。基础步骤：$P(2)$为真，因为$(p_1 \to p_2) \to (p_1 \to p_2)$是重言式。归纳步骤：假设$P(n)$为真。为了证明$[(p_1 \to p_2) \land (p_2 \to p_3) \land \cdots \land (p_{n-1} \to p_n) \land (p_n \to p_{n+1})] \to [(p_1 \land p_2 \land \cdots \land p_{n-1} \land p_n) \to p_{n+1}]$是重言式，假设这个蕴涵式的前件为真。因为前件和$P(n)$都为真，所以$(p_1 \land p_2 \land \cdots \land p_{n-1}) \to p_n$为真。因为这个式子为真，并且因为$p_n \to p_{n+1}$(它是假设的一部分)，所以根据假言三段论，$(p_1 \land p_2 \land \cdots \land p_{n-1}) \to p_{n+1}$为真。从这个式子得出较弱的命题$(p_1 \land p_2 \land \cdots \land p_{n-1} \land p_n) \to p_{n+1}$。

47. 若$n+1=2$，则这两个集合不重叠。事实上，蕴涵式$P(1) \to P(2)$为假。

49. 假设良序性成立。假设$P(1)$为真且对每一个正整数$n \geq 1$来说，$(P(1) \land P(2) \land \cdots \land P(n)) \to P(n+1)$为真。设$S$是$P(n)$为假的整数$n$的集合，将证明$S=\emptyset$。假设$S \neq \emptyset$。则根据良序性，在$S$里存在最小整数$m$。我们知道$m$不能是1，因为$P(1)$为真。因为$n=m$是使得$P(n)$为假的最小整数，所以$P(1), P(2), \cdots, P(m-1)$都为真，并且$m-1 \geq 1$。因为$(P(1) \land P(2) \land \cdots \land P(m-1)) \to P(m)$为真，所以$P(m)$必然为真，矛盾。因此$S=\phi$。

51. 设$P(n)$是"$H_{2^n} \leq 1$"。基础步骤：$P(0)$为真，因为$H_{2^0} = H_1 = 1 \leq 1+0$。归纳步骤：假设$H_{2^n} \leq 1+n$，则

$$H_{2^{n+1}} = H_{2^n} + \sum_{j=2^n+1}^{2^{n+1}} \frac{1}{j} \leq 1 + n + 2^n(\frac{1}{2^{n+1}}) < 1 + n + 1 = 1 + (n+1)$$

53. 设$P(n)$是"$1/\sqrt{1} + 1/\sqrt{2} + 1/\sqrt{3} + \cdots + 1/\sqrt{n} > 2(\sqrt{n+1} - 1)$"。基础步骤：$P(1)$为真，因为$1 > 2(\sqrt{2} - 1)$。归纳步骤：假设$P(n)$为真。则$1/\sqrt{1} + 1/\sqrt{2} + 1/\sqrt{3} + \cdots + 1/\sqrt{n} + 1/\sqrt{n+1} > 2(\sqrt{n+1} - 1) + 1/\sqrt{n+1}$。若证明$2(\sqrt{n+1} - 1) + 1/\sqrt{n+1} > 2(\sqrt{n+2} - 1)$，则得出$P(n+1)$为真。这个不等式等价于$2(\sqrt{n+2} - \sqrt{n+1}) < 1/\sqrt{n+1}$，它等价于$2(\sqrt{n+2} - \sqrt{n+1})(\sqrt{n+2} + \sqrt{n+1}) < \sqrt{n+1}/\sqrt{n+1} + \sqrt{n+2}/\sqrt{n+1}$。这等价于$2 < 1 + \sqrt{n+2}/\sqrt{n+1}$，它显然为真。

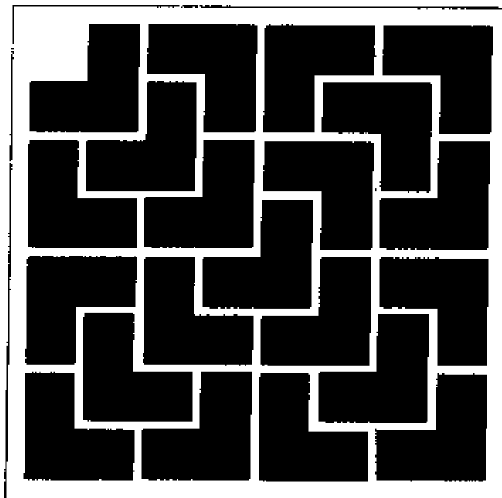
55. 将首先证明当n是2的幂时的结果，即若$n=2^k, k=1,2,\cdots$。设$P(k)$是命题：$A \geq G$，其中A和G是一组$n=2^k$个正实数的算术和几何平均值。基础步骤：$k=1$并且$n=2^1=2$。注意$(\sqrt{a_1} - \sqrt{a_2})^2 \geq 0$。展开这个式子证明$a_1 - 2\sqrt{a_1}\sqrt{a_2} + a_2 \geq 0$，即$(a_1 + a_2)/2 \geq (a_1 a_2)^{1/2}$。归纳步骤：假设$P(k)$为真，并且$n=2^k$。我们将证明$P(k+1)$为真。有$2^{k+1} = 2n$，则$(a_1 + a_2 + \cdots + a_{2n})/(2n) = ((a_1 + a_2 + \cdots + a_n)/n + (a_{n+1} + a_{n+2} + \cdots + a_{2n})/n)/2$。同理，$(a_1 a_2 \cdots a_{2n})^{1/(2n)} = [(a_1 a_2 \cdots a_n)^{1/n} (a_{n+1} a_{n+2} \cdots a_{2n})^{1/n}]^{1/2}$。为了简化记号，设$A(x, y, \cdots)$和$G(x, y, \cdots)$分别表示$x, y, \cdots$的算术平均值和几何平均值。另外，若$x \leq x'$，$y \leq y'$，依次类推，则$A(x, y, \cdots) \leq A(x', y', \cdots)$且$G(x, y, \cdots) \leq$

$G(x',y',\cdots)$。因此，$A(a_1,a_2,\cdots,a_{2n})=A(A(a_1,a_2,\cdots,a_n),A(a_{n+1},a_{n+2},\cdots,a_{2n}))\geqslant A(G(a_1,a_2,\cdots,a_n),G(a_{n+1},a_{n+2},\cdots,a_{2n}))\geqslant G(G(a_1,a_2,\cdots,a_n),G(a_{n+1},a_{n+2},\cdots,a_{2n}))=G(a_1,a_2,\cdots,a_{2n})$。这样就完成了对 2 的幂的情形的证明。现在若 n 不是 2 的幂，则设 m 是 2 的下一个更高的幂，并且设 a_{n+1}，…，a_m 都等于$A(a_1,a_2,\cdots,a_n)=\bar{a}$。于是就有$((a_1a_2\cdots a_n)\bar{a})^{m-n})^{1/m}\leqslant A(a_1,a_2,\cdots,a_m)$，因为 m 是 2 的幂。因为$A(a_1,a_2,\cdots,a_m)=\bar{a}$，所以$(a_1a_2\cdots a_n)^{1/m}\bar{a}^{1-n/m}\leqslant\bar{a})^{n/m}$。两边都求第 m/n 次幂就得出$G(a_1,a_2,\cdots,a_n)\leqslant A(a_1,a_2,\cdots,a_n)$。

57. 当 $n=1$ 时，对基础情形来说，没有什么需要证明的。现在假设归纳假设成立。假设 $p\mid a_1a_2\cdots a_na_{n+1}$。注意 $\gcd(p,a_1,a_2,\cdots,a_n)=1$ 或 p。若它是 1，则根据 2.5 节引理 1，有 $p\mid a_{n+1}$。若它是 p，则 $p\mid a_1a_2\cdots a_n$，所以根据归纳假设，对某个 $i\leqslant n$ 来说有 $p\mid a_i$。这样就完成了证明。

59. 设$P(n)$是命题：若 x_1，x_2，…，x_n 是 n 个不同的实数，则无论在乘积里如何插入括号，都需要用 $n-1$ 次乘法来求出这些数的乘积。将用数学归纳法第二原理来证明$P(n)$为真。基础情形 P（1）为真，因为需要 $1-1=0$ 次乘法来求出 x_1 的乘积，即只有一个因子的乘积。假设对 $1\leqslant k\leqslant n$ 来说 P（k）为真。求出 $n+1$ 个不同实数 x_1，x_2，…，x_n，x_{n+1}的乘积所用的最后一次乘法，是对某个 k 来说这些数的前k 个数的乘积与后 $n+1-k$ 个数的乘积之间的乘法。根据归纳假设，需要用 $k-1$ 次乘法来求出这些数中 k 个数的乘积，无论在这些数的乘积里如何插入括号；而且需要用 $n-k$ 次乘法来求出这些数中其余$n+1-k$ 个数的乘积，也无论在这些数的乘积里如何插入括号。因为需要另外一次乘法来求出所有这些 $n+1$ 个数的乘积，所以使用的乘法总次数等于$(k-1)+(n-k)+1=n$。因此$P(n+1)$为真。证毕。

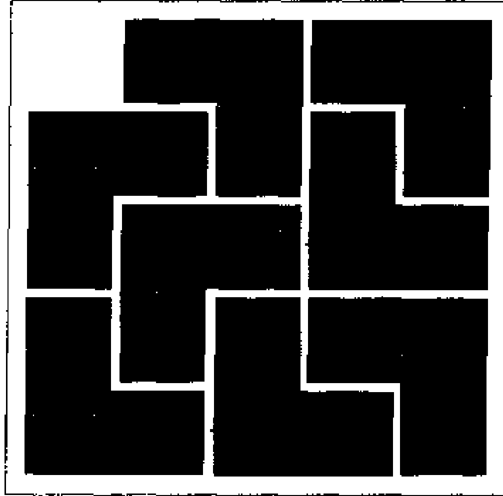
61.



63. 设$P(n)$是命题：每一个去掉了一个 1×1×1 立方体的 $2^n\times2^n\times2^n$ 棋盘，都可以用每个都去掉了一个 1×1×1 立方体的 2×2×2 立方体砖块来铺满。基础步骤$P(1)$成立，因为一个砖块与所需要铺满的立体是重合的。假设$P(n)$为真，现在考虑去掉了一个 1×1×1 立方体的 $2^{n+1}\times2^{n+1}\times2^{n+1}$立方体目标。用经过目标中心并且与目标表面平行的平面来把这个目标分成八块。所去掉的 1×1×1 块出现在这八块当中的一块里。现在放一块砖，让它的中心是在这个大目标的中心上，使得它所去掉的 1×1×1 立方体是在大目标

去掉一个 1×1×1 立方体的象限里。这样就产生出八个 $2^n \times 2^n \times 2^n$ 立方体，每个都去掉一个 1×1×1 立方体。根据归纳假设，可以用砖块铺满这八个目标当中的每一个。把这些铺砖系统放在一起就产生出所需要的铺砖系统。

65.



67. 假定 $a = dq + r = dq' + r'$ 使得 $0 \leqslant r < d$ 并且 $0 \leqslant r' < d$，则 $d(q - q') = r' - r$。由此得出 d 整除 $r' - r$。因为 $-d < r' - r < d$，所以 $r' - r = 0$。因此，$r' = r$。由此得出 $q = q'$。

69. 这里自引用所引起的悖论，答案显然是"否"。只有有穷多个英语单词，所以只有有穷多个包含 15 个或更少单词的字符串；因此只有有穷多个正整数可以如此描述，而不是所有的正整数。

3.3 节

1. a) $f(1)=3, f(2)=5, f(3)=7, f(4)=9$
   b) $f(1)=3, f(2)=9, f(3)=27, f(4)=81$
   c) $f(1)=2, f(2)=4, f(3)=16, f(4)=65\ 536$
   d) $f(1)=3, f(2)=13, f(3)=183, f(4)=33\ 673$
3. a) $f(2)=-1, f(3)=5, f(4)=2, f(5)=17$
   b) $f(2)=-4, f(3)=32, f(4)=-4096, f(5)=536\ 870\ 912$
   c) $f(2)=8, f(3)=176, f(4)=92\ 672, f(5)=25\ 764\ 174\ 848$
   d) $f(2)=-1/2, f(3)=-4, f(4)=1/8, f(5)=-32$
5. 有许多种可能正确的答案，我们将提供相对较简单的答案。
   a) 对 $n \geqslant 1$ 来说 $a_{n+1} = a_n + 6$ 并且 $a_1 = 6$
   b) 对 $n \geqslant 1$ 来说 $a_{n+1} = a_n + 2$ 并且 $a_1 = 3$
   c) 对 $n \geqslant 1$ 来说 $a_{n+1} = 10a_n$ 并且 $a_1 = 10$
   d) 对 $n \geqslant 1$ 来说 $a_{n+1} = a_n$ 并且 $a_1 = 5$
7. $F(0)=0$，对 $n \geqslant 1$ 来说 $F(n) = F(n-1) + n$
9. $P_m(0)=0, P_m(n+1) = P_m(n) + m$
11. 设 $P(n)$ 是"$f_1 + f_3 + \cdots + f_{2n-1} = f_{2n}$"。基础步骤：$P(1)$ 为真，因为 $f_1 = 1 = f_2$。归纳步骤：假设 $P(n)$ 为真，则 $f_1 + f_3 + \cdots + f_{2n-1} + f_{2n+1} = f_{2n} + f_{2n+1} = f_{2n+2} + f_{2(n+1)}$。

13. 基础步骤：$f_0f_1+f_1f_2=0\cdot1+1\cdot1=1^2=f_2^2$。归纳步骤：假设 $f_0f_1+f_1f_2+\cdots+f_{2n-1}f_{2n}=f_{2n}^2$，则 $f_0f_1+f_1f_2+\cdots+f_{2n-1}f_{2n}+f_{2n}f_{2n+1}+f_{2n+1}f_{2n+2}=f_{2n}^2+f_{2n}f_{2n+1}+f_{2n+1}f_{2n+2}=f_{2n}(f_{2n}+f_{2n+1})+f_{2n+1}f_{2n+2}=f_{2n}f_{2n+2}+f_{2n+1}f_{2n+2}=(f_{2n}+f_{2n+1})f_{2n+2}=f_{2n+2}^2$。

15. 欧几里得算法为求出gcd(f_{n+1},f_n)而使用的除法次数对 $n=0$ 来说是 0，对 $n=1$ 来说是 1，而对 $n\geqslant2$ 来说是 $n-1$。为了对 $n\geqslant2$ 证明这个结果，用数学归纳法。对 $n=2$ 来说，一次除法后有gcd$(f_3,f_2)=$gcd$(2,1)=$gcd$(1,0)=1$ 。现在假设用 $n-1$ 次除法来求出gcd(f_{n+1},f_n)。为了求出gcd(f_{n+2},f_{n+1})，首先让 f_{n+2}除以 f_{n+1}来得出 $f_{n+2}=1\cdot f_{n+1}+f_n$。在一次除法之后有gcd$(f_{n+2},f_{n+1})=gcd(f_{n+1},f_n)$。根据归纳假设，所以恰好需要另外 $n-1$ 次除法。这说明需要 n 次除法来求出gcd(f_{n+2},f_{n+1})。证毕。

17. $|A|=-1$。因此$|A^n|=(-1)^n$。所以 $f_{n+1}f_{n-1}-f_n^2=(-1)^n$。

19. a) 用归纳法来证明。基础步骤：对 $n=1$ 来说，$\max(-a_1)=-a_1=-\min(a_1)$。对 $n=2$ 来说，有两种情形。若 $a_2\geqslant a_1$，则 $-a_1\geqslant-a_2$，所以$\max(-a_1,-a_2)=-a_1=-\min(a_1,a_2)$。若 $a_2<a_1$，则 $-a_1<-a_2$，所以$\max(-a_1,-a_2)=-a_2=-\min(a_1,a_2)$。归纳步骤：假设对满足 $n\geqslant2$ 的 n 来说命题为真，则$\max(-a_1,-a_2,\cdots,-a_n,-a_{n+1})=\max(\max(-a_1,-a_2,\cdots,-a_n),-a_{n+1})=\max(-\min(a_1,a_2,\cdots,a_n),-a_{n+1})=-\min(\min(a_1,a_2,\cdots,a_n),a_{n+1})=-\min(a_1,\cdots,a_{n+1})$。

    b) 用数学归纳法来证明。对 $n=1$ 来说，结果是恒等式 $a_1+b_1=a_1+b_1$。对 $n=2$ 来说，首先考虑 $a_1+b_1\geqslant a_2+b_2$ 的情形。则$\max(a_1+b_1,a_2+b_2)=a_1+b_1$ 。另外注意 $a_1\leqslant\max(a_1,a_2)$和 $b_1\leqslant\max(b_1,b_2)$，所以$a_1+b_1\leqslant\max(a_1,a_2)+\max(b_1,b_2)$。因此 $\max(a_1+b_1,a_2+b_2)=a_1+b_1\leqslant\max(a_1,a_2)+\max(b_1,b_2)$。$a_1+b_1<a_2+b_2$ 的情形是类似的。对于归纳步骤，假设对 n 来说结果为真，则$\max(a_1+b_1,a_2+b_2,\cdots,a_n+b_n,a_{n+1}+b_{n+1})=\max(\max(a_1+b_1,a_2+b_2,\cdots,a_n+b_n),a_{n+1}+b_{n+1})\leqslant\max(\max(a_1,a_2,\cdots,a_n)+\max(b_1,+b_2,\cdots,b_n),a_{n+1}+b_{n+1})\leqslant\max(\max(a_1,a_2,\cdots,a_n),a_{n+1})+\max(\max(b_1,b_2,\cdots,b_n),b_{n+1})=\max(a_1,a_2,\cdots,a_n,a_{n+1})+\max(b_1,b_2,\cdots,b_n,b_{n+1})$。

    c) 与 b) 一样，但是把 “max” 的每一次出现都换成 “min”，并且反转每个不等式的方向。

21. $5\in S$，并且若 x，$y\in S$，则 $x+y\in S$。

23. a) $0\in S$，并且若 $x\in S$，则 $x+2\in S$ 和 $x-2\in S$。

    b) $2\in S$，并且若 $x\in S$，则 $x+3\in S$。

    c) $1\in S$，$2\in S$，$3\in S$，$4\in S$，并且若 $x\in S$，则 $x+5\in S$。

25. 若 x 是集合或是表示集合的变元，则 x 是合式公式。若 x 和 y 都是合式公式，则$\overline{x}$，$(x\cup y)$，$(x\cap y)$和$(x-y)$都是合式公式。

27. $\lambda^R=\lambda$ 并且对 $x\in\Sigma$，$u\in\Sigma^*$来说 $(ux)^R=xu^R$。

29. $w^0=\lambda$ 并且 $w^{n+1}=ww^n$。

31. 当对某个非负整数 n 来说，该字符串是由 n 个 0 后跟 n 个 1 所组成的时。

33. 设 $P(i)$ 是 “$l(w^i)=i\cdot l(w)$”。$P(0)$为真，因为 $l(w^0)=0=0\cdot l(w)$。假设 $P(i)$为真。则 $l(w^{i+1})=l(ww^i)=l(w)+l(w^i)=l(w)+i\cdot l(w)=(i+1)\cdot l(w)$。

35. a) $P_{m,m}=P_m$，因为在 m 的分拆里不可能使用比 m 大的数。
    b) 因为只有一种方式来分拆 1，即 $1=1$，所以 $P_{1,n}=1$。因为只有一种方式来把 m 分拆成 1，所以 $P_{m,1}=1$。当 $n>m$ 时，得出 $P_{m,n}=P(m,m)$，因为不可能使用比 m 大的数。$P_{m,m}=1+P_{m,m-1}$，因为当在分拆里允许使用 m 时，出现另外一个分拆，即 $m=m$。若 $m>n$，则 $P_{m,n}=P_{m,n-1}+P_{m-n,n}$，因为一个把 m 分成不大于 n 的整数的分拆，要么不使用任何 n，因此算在 $P_{m,n-1}$里；要么使用一个 n 和一个对 $m-n$ 的分拆，因此算在 $P_{m-n,n}$里。
    c) $P_5=7$，$P_6=11$
37. 设 $P(n)$是"$A(n,1)=4$"。基础步骤：$P(1)$为真，因为$A(1,2)=A(0,A(1,1))=A(0,2)=2\cdot 2=4$。归纳步骤：假设$P(n)$为真，即$A(n,2)=4$，则$A(n+1,2)=A(n,A(n+1,1))=A(n,2)=4$。
39. a) 16　b) 65 536
41. 用双重归纳论证来证明更强的命题：当 $k>l$ 时，$A(m,k)>A(m,l)$。基础步骤：当 $m=0$ 时命题为真，因为 $k>l$ 蕴涵$A(0<k)=2k>2l=A(0,l)$。归纳步骤：假设对所有满足 $x>y$ 的非负整数x 和y 来说都有$A(m,x)>A(m,y)$。将证明这个假设蕴涵着：若 $k>l$，则$A(m+1,k)>A(m+1,l)$。基础步骤：当 $l=0$ 而且 $k>0$ 时，$A(m+1,l)=0$，并且或者$A(m+1,k)=2$，或者$A(m+1,k)=A(m,A(m+1,k-1))$。若 $m=0$,则它是$2A(1,k-1)=2^k$。若 $m>0$，则根据归纳假设，它是大于 0 的。在所有情形里，$A(m+1,k)>0$，事实上，$A(m+1,k)\geq 2$。若 $l=1$ 而且 $k>0$，$A(m+1,l)=2$，并且$A(m+1,k)=A(m,A(m+1,k-1))$其中，$A(m+1,k-1)\geq 2$。因此根据归纳假设，$A(m,A(m+1,k-1))\geq A(m,2)>A(m,1)=2$。归纳步骤：假设对所有$r>s(s=0,1,\cdots,l)$来说，都有$A(m+1,r)>A(m+1,s)$。于是若 $k+1>l+1$，则得出$A(m+1,k+1)=A(m,A(m+1,k))>A(m,A(m+1,k))=A(m+1,l+1)$。
43. 从练习 42 得出$A(i,j)\geq A(i-1,j)\geq\cdots\geq A(0,j)=2j\geq j$。
45. 设$P(n)$是"$F(n)$是良定义的"。则$P(0)$为真，因为规定了$F(0)$。假设对所有 $k<n$ 来说$P(k)$为真。则在 n 上$F(n)$是良定义的，因为$F(n)$是利用$F(0)$，$F(1)$，…，$F(n-1)$来给出的。所以对所有整数 n 来说$P(n)$为真。
47. a) $F(1)$的值是二义性的。
    b) $F(2)$是无定义的，因为$F(0)$是无定义的。
    c) $F(3)$是二义性的并且$F(4)$是无定义的，因为$F(\frac{4}{3})$没有意义。
    d) $F(1)$的定义是二义性的，因为第 2 和第 3 个句子似乎都可以应用。
    e) 不能计算$F(2)$，因为要计算$F(2)$，结果会给出$F(2)=1+F(F(1))=1+F(2)$。
49. a) 1　b) 2　c) 3　d) 3　e) 4　f) 4　g) 5
51. $f_0^*(n)=\lceil n/a\rceil$
53. 对 $n\geq 2$ 来说，$f_0^*(n)=\lceil loglogn\rceil$，$f_0^*$ (1) $=0$
55. 将证明$a(n)$是自然数并且$a(n)\leq n$。对基础情形 $n=0$ 来说，这个命题为真，因为$a(0)=0$。现在假设$a(n-1)$是自然数并且$a(n-1)\leq n-1$。则$a(a(n-1))$是 a 应用到小于或等于 $n-1$ 的自然数上。因此$a(a(n-1))$也是小于或等于 $n-1$ 的自然数。因此$n-a(a(n-1))$是 n 减去某个小于或等于 $n-1$ 的自然数，结果是小于或等于 n 的自然数。

57. 根据练习 56，有 $a(n)=\lfloor(n+1)\mu\rfloor$ 和 $a(n-1)=\lfloor n\mu\rfloor$。因为 $\mu<1$，所以这两个值相等或相差 1。首先假设 $\mu m-\lfloor\mu n\rfloor<1-\mu$。这等价于 $\mu(n+1)<1+\lfloor\mu n\rfloor$。若这个假设为真，则 $\lfloor\mu(n+1)\rfloor=\lfloor\mu n\rfloor$。在另一方面，若 $\mu m-\lfloor\mu n\rfloor\geq 1-\mu$，则 $\mu(n+1)\geq 1+\lfloor\mu n\rfloor$，所以 $\lfloor\mu(n+1)\rfloor=\lfloor\mu n\rfloor+1$，即为所求。

59. $f(0)=1, m(0)=0; f(1)=1, m(1)=0; f(2)=2, m(2)=1; f(3)=2, m(3)=2; f(4)=3, m(4)=2, f(5)=3, m(5)=3; f(6)=4, m(6)=4; f(7)=5, m(7)=4; f(8)=5, m(8)=5; f(9)=6, m(9)=6$。

61. n 的最后一次出现是在这样的位置上：1，2，…，n 的总数求和就是那个位置数。因为 a_k 是 k 的出现次数，所以这个数就是 $\sum_{k=1}^{n} a_k$，即为所求。因为 $f(n)$ 是这个序列的前 n 项之和，所以 $f(f(n))$ 是这个序列的前 $f(n)$ 项之和。因为 $f(n)$ 是值为 n 的最后一项，这意味着这个和是这个序列里值不超过 n 的所有项之和。因为这个序列里值为 k 的项有 a_k 个，所以这个和是 $\sum_{k=1}^{n} k\cdot a_k$，即为所求。

63. 假设 $f(n)$ 和 $g(n)$ 都是属于 $\mathscr{L}$ 的并且都是逐渐正的。则 $\ln f(n)$ 和 $\ln g(n)$ 都是属于 $\mathscr{L}$ 的，所以根据练习 62，$\ln f(n)+\ln g(n)=\ln(f(n)g(n))\in\mathscr{L}$。因此，$e^{\ln(f(n)g(n))}=f(n)g(n)\in\mathscr{L}$。同理，$\ln f(n)-\ln g(n)=\ln(f(n)/g(n))\in\mathscr{L}$。所以 $e^{\ln(f(n)/g(n))}=f(n)/g(n)\in\mathscr{L}$。

现在假设 $f(n)$ 和 $g(n)$ 都是属于 $\mathscr{L}$ 的并且都不恒等于 0。若 $f(n)$ 和 $g(n)$ 都是逐渐负的，则 $-f(n)$ 和 $-g(n)$ 都是逐渐正的并且都是属于 $\mathscr{L}$ 的，所以 $f(n)g(n)=(-f(n))\cdot(-g(n))$ 和 $f(n)/g(n)=(-f(n))/(-g(n))$ 都是属于 $\mathscr{L}$ 的。若这两个函数中的一个（比如说 $f(n)$）是逐渐正的，而另一个是逐渐负的，则 $f(n)g(n)=(-f(n))(-g(n))\in\mathscr{L}$；商的情况类似。

65. 若 $f(n)\in\mathscr{L}$ 并且是逐渐正的，则 $\ln f(n)\in\mathscr{L}$，所以根据练习 63 的结果，$\frac{1}{2}\ln f(n)=\ln(\sqrt{f(n)})\in\mathscr{L}$。因此 $\sqrt{f(n)}=e^{\ln\sqrt{f(n)}}\in\mathscr{L}$。

3.4 节

1. **procedure** *mult*(n:正整数，x:整数)
   if $n=1$ **then** $mult(n,x):=x$
   else $mult(n,x):=x+mult(n-1,x)$
3. **procedure** *sum of odds*(n:正整数)
   if $n=1$ **then** $sum\ of\ odds(n):=1$
   else $sum\ of\ odds(n):=sum\ of\ odds(n-1)+2n-1$
5. **procedure** *smallest*($a_1,\cdots,a_n$:整数)
   if $n=1$ **then** $smallest(a_1,\cdots,a_n):=a_1$
   else $smallest(a_1,\cdots,a_n):=\min(smallest(a_1,\cdots,a_{n-1}),a_n)$
7. **procedure** *modfactorial*(n,m:正整数)
   if $n=1$ **then** $modfactorial(n,m):=1$
   else $modfactorial(n,m):=(n * modfactoial(n-1,m))\bmod m$
9. **procedure** *gcd*(a,b:非负整数)
   {假设 $a<b$ 是成立的}

```
if a = 0 then gcd(a, b) := b
else if a = b - a then gcd(a, b) := a
else if a < b - a then gcd(a, b) := gcd(a, b - a)
else gcd(a, b) := gcd(b - a, a)
```

11. n 次乘法与 2^n 次乘法

13. $O(\log n)$与 n

15.
```
procedure a(n:非负整数)
  if n = 0 then a(n) := 1
  else if n = 1 then a(n) := 2
  else a(n) := a(n - 1) * a(n - 2)
```

17. 迭代

19.
```
procedure iterative(n:非负整数)
  if n = 0 then z := 1
  else if n = 1 then z := 2
  else
  begin
    x := 1
    y := 2
    z := 3
    for i := 1 to n - 2
      begin
        w := x + y + z
        x := y
        y := z
        z := w
      end
  end
  {z 是该序列的第 n 项}
```

21. 首先给出递归过程，然后给出迭代过程。

```
procedure r(n:非负整数)
if n < 3 then r(n) := 2n + 1
else r(n) := r(n - 1)·(r(n - 2))^2·(r(n - 3))^3
procedure i(n:非负整数)
if n = 0 then z := 1
else if n = 1 then z := 3
else
begin
  x := 1
  y := 3
  z := 5
```

```
      for i := 1 to n - 2
      begin
          w := z * y^2 * x^3
          x := y
          y := z
          z := w
      end
end
```

{z 是该序列的第 n 项}

这个迭代过程是更高效率的。

23.
```
procedure reverse(w: 位串)
n := length(w)
if n ≤ 1 then reverse(w) := w
else reverse(w) := substr(w, n, n) reverse(substr(w, 1, n - 1))
```
{substr (w, a, b) 是包括从第 a 个位置到第 b 个位置的 w 的子串}

25.
```
procedure A(m, n: 非负整数)
if m = 0 then A(m, n) := 2n
else if n = 0 then A(m, n) := 0
else if n = 1 then A(m, n) := 2
else A(m, n) := A(m - 1, A(m, n - 1))
```

3.5 节

1. 假设 $x=0$。这个程序段首先对 y 赋值 1 然后对 z 赋值 $x+y=0+1=1$。
3. 假设 $y=3$。这个程序段首先对 x 赋值 2 然后对 z 赋值 $x+y=2+3=5$。因为 $y=3>0$，所以它对 z 赋值 $z+1=5+1=6$。
5. $(p \wedge condition)\{S_1\}q$

   $(p \wedge \neg condition1 \wedge condition2)\{S_2\}q$

   $\vdots$

   $(p \wedge \neg condition1 \wedge \neg condition2 \cdots \wedge \neg condition(n-1))\{S_n\}q$

   $\therefore p\{$**if** $condition1$ **then** S_1; **else if** $condition2$ **then** S_2; $\cdots$; **else** $S_n\}\ q$
7. 将证明 p：“$power = x^{i-1}$并且 $i \leqslant n+1$” 是循环不变量。注意起初 p 为真，因为在循环开始前，$i=1$ 并且 $power = x^0 = x^{i-1}$。其次必须证明在循环的一遍执行后，若 p 为真并且 $i \leqslant n$，则在循环的下一遍执行后，p 仍然为真。循环把 i 增加 1。因此在这遍循环之前 $i \leqslant n$，在这遍循环之后 $i \leqslant n+1$。而且该循环对 $power$ 赋值 $power \cdot x$。根据归纳假设，$power$ 赋值为 $x^{i-1} \cdot x = x^i$。因此 p 仍然为真。另外，在该循环执行 n 遍之后，循环结束并且 $i=n+1$，因为在进入循环前把 i 赋值为 1，在每遍循环里把 i 增加 1，并且当 $i>n$ 时循环结束。所以，在结束时 $power = x^n$，正如所需。
9. 假设 p 是 “m 和 n 都是整数”。于是若条件 $n<0$ 为真，则在执行 S_1 之后有 $a=-n=|n|$。若条件 $n<0$ 为假，则在执行 S_1 之后有 $a=n=|n|$。因此$p\{S_1\}q$ 为真，其中 q 是 $p \wedge (a=|n|)$。因为 S_2 赋值 0 给 k 和 x，显然$q\{S_2\}r$ 为真，其中 r 是$q \wedge (k=0) \wedge (x$

$=0$)。假设 r 为真。设 $P(k)$ 是"$x=mk$ 并且 $k\leqslant a$"。可以证明对 S_3 里的循环来说 $P(k)$ 是循环不变量。$P(0)$ 为真，因为在进入循环前 $x=0=m\cdot 0$ 并且 $0\leqslant a$。现在假设 $P(k)$ 为真并且 $k<a$。则 $P(k+1)$ 为真，因为把 x 赋值为 $x+m=mk+m=m(k+1)$。当 $k=a$ 时循环结束，而在这个时候 $x=ma$。因此 $r\{S_3\}s$ 为真，其中 s 是"$a=|n|$ 并且 $x=ma$"。现在假设 s 为真。于是若 $n<0$，则 $a=-n$，所以 $x=-mn$。在这种情形里，S_4 给 *product* 赋值 $-x=mn$。若 $n>0$，则 $x=ma=mn$，所以 S_4 给 *product* 赋值 mn。因此 $s\{S_4\}t$ 为真。

11. 假设初始断言 p 为真。则因为 $p\{S\}q_0$ 为真，所以在执行程序段 S 后 q_0 为真。因为 $q_0\rightarrow q_1$ 为真，所以得出在执行程序段 S 后 q_1 为真。因此 $p\{S\}q_1$ 为真。
13. 将用命题 p："$\gcd(a,b)=\gcd(x,y)$ 并且 $y\geqslant 0$"来作为循环不变量。注意在进入循环前 p 为真，因为使用初始断言在进入循环时 $x=a$，$y=b$，且 y 是正整数。现在假设 p 为真并且 $y>0$；则循环将再度执行。在循环内部，x 和 y 分别换成 y 和 $x \bmod y$。根据 2.4 节的引理 1，$\gcd(x,y)=\gcd(y,x \bmod y)$。因此，在循环执行后，$\gcd(x,y)$ 的值与循环执行前是一样的。另外，因为 y 是余数，所以它至少为 0。因此 p 仍然为真，所以它是循环不变量。另外，若循环结束，则 $y=0$。在这种情形里，有 $\gcd(x,y)=x$，即终结断言。因此这个以 x 为输出的程序正确地计算了 $\gcd(a,b)$。最后，可以证明循环必然结束，因为每次迭代导致 y 的值至少减少 1。因此，循环至多可以迭代 b 次。

补充练习

1. 设 $a=2n+1$ 和 $b=2m+1$。则 $ab=(2n+1)(2m+1)=2(2nm+m+n)+1$，它是奇数。
3. 假。$\sqrt{2}+(-\sqrt{2})=0$ 是个反例。
5. 这是肯定结论谬误的例子。
7. 分情形证明。情形 1：$x\geqslant 0$ 且 $y\geqslant 0$。则 $|xy|=xy=|x||y|$。情形 2：$x\geqslant 0$ 且 $y<0$。则 $|xy|=-xy=x(-y)=|x||y|$。情形 3：$x<0$ 且 $y\geqslant 0$。则 $|xy|=-xy=(-x)y=|x||y|$。情形 4：$x<0$ 且 $y<0$。则 $|xy|=xy=(-x)(-y)=|x||y|$。
9. 假设 x_j 各不相同。设 $P(x)$ 与提示里的一样，则 $P(x)$ 是多项式（事实上为 $n-1$ 次）；而且若 $x=x_m$，则 $\Pi_{i\neq j}(x-x_j)/(x_i-x_j)=0$，除非 $i=m$。因此 $P(x_m)=\Pi_{j\neq m}y_m(x_m-x_j)/(x_m-x_j)=1\cdot y_m=y_m$。
11. 设 $P(n)$ 是"$1\cdot 1+2\cdot 2+\cdots+n\cdot 2^{n-1}=(n-1)2^n+1$"。基础步骤：$P(1)$ 为真，因为 $1\cdot 1=1=(1-1)2^1+1$。归纳步骤：假设 $P(n)$ 为真，则 $1\cdot 1+2\cdot 2+\cdots+n\cdot 2^{n-1}+(n+1)\cdot 2^n=(n-1)2^n+1+(n+1)2^n=2n\cdot 2^n+1=((n+1)-1)2^{n+1}+1$。
13. 设 $P(n)$ 是"$1/(1\cdot 4)+\cdots+1/[(3n-2)(3n+1)]=n/(3n+1)$"。基础步骤：$P(1)$ 为真，因为 $1/(1\cdot 4)=1/4$。归纳步骤：假设 $P(n)$ 为真，则 $1/(1\cdot 4)+\cdots+1/[(3n-2)(3n+1)]+1/[(3n+1)(3n+4)]=n/(3n+1)+1/[(3n+1)(3n+4)]=[n(3n+4)+1]/[(3n+1)(3n+4)]=[(3n+1)(n+1)]/[(3n+1)(3n+4)]=(n+1)/(3n+4)$。
15. 设 $P(n)$ 是"$2^n>n^3$"。基础步骤：$P(10)$ 为真，因为 $1024>1000$。归纳步骤：假设 $P(n)$ 为真，则 $(n+1)^3=n^3+3n^2+3n+1\leqslant n^3+9n^2\leqslant n^3+n^3=2n^3<2\cdot 2^n=2^{n+1}$。
17. 设 $P(n)$ 是"$a-b$ 是 a^n-b^n 的因子"。基础步骤：$P(1)$ 平凡地为真。归纳步骤：假设 $P(n)$ 为真，则 $a^{n+1}-b^{n+1}=a^{n+1}-ab^n+a^{n+1}-b^{n+1}=a(a^n-b^n)+b^n(a-b)$。因为

$a-b$ 是 a^n-b^n 的因子，且 $a-b$ 是 $a-b$ 的因子，所以 $a-b$ 是 $a^{n+1}-b^{n+1}$ 的因子。

19. 设 $P(n)$ 是"$a+(a+d)+\cdots+(a+nd)=(n+1)(2a+nd)/2$"。基础步骤：$P(1)$ 为真，因为 $a+(a+d)=2a+d=2(2a+d)/2$。归纳步骤：假设 $P(n)$ 为真，则 $a+(a+d)+\cdots+(a+nd)+(a+(n+1)d)=(n+1)(2a+nd)/2+a+(n+1)d=\frac{1}{2}[2an+2a+n^2d+nd+2a+2nd+2d]=\frac{1}{2}[2an+4a+n^2d+3nd+2d]=\frac{1}{2}(n+2)(2a+(n+1)d)$。

21. 将用数学归纳法第二原理来证明：若 $n\equiv 0 \pmod 3$，则 f_n 为偶数；否则 f_n 为奇数。基础步骤成立，因为 $f_0=0$ 是偶数而 $f_1=1$ 是奇数。现在假设如果 $k\leqslant n$，那么若 $k\equiv 0 \pmod 3$，则 f_k 为偶数；否则 f_k 为奇数。现在假设 $n+1\equiv 0 \pmod 3$，则 $f_{n+1}=f_n+f_{n-1}$ 是偶数，因为 f_n 和 f_{n-1} 都是奇数。若 $n+1\equiv 1 (\mathrm{mod})3$，则 $f_{n+1}=f_n+f_{n-1}$ 是奇数，因为 f_n 是偶数而 f_{n-1} 是奇数。最后，若 $n+1\equiv 2 \pmod 3$，则 $f_{n+1}=f_n+f_{n-1}$ 是奇数，因为 f_n 是奇数而 f_{n-1} 是偶数。这样就完成了归纳证明。

23. 设 $P(n)$ 是命题：对每个非负整数 k 来说，有 $f_kf_n+f_{k+1}f_{n+1}=f_{n+k+1}$。基础步骤包括证明 $P(0)$ 和 $P(1)$ 都成立。$P(0)$ 为真，因为 $f_kf_0+f_{k+1}f_1=f_{k+1}\cdot 0+f_{k+1}\cdot 1=f_1$。因为 $f_kf_1+f_{k+1}f_2=f_k+f_{k+1}=f_{k+2}$，所以 $P(1)$ 为真。现在假设 $P(n)$ 成立。则根据归纳假设和斐波那契数的递归定义，得出 $f_{k+1}f_{n+1}+f_{k+2}f_{n+2}=f_k(f_{n-1}+f_n)+f_{k+1}(f_n+f_{n+1})=(f_kf_{n-1}+f_{k+1}f_n)+(f_kf_n+f_{k+1}f_{n+1})=f_{n-1+k+1}+f_{n+k+1}=f_{n+k+2}$。这样就证明了 $P\ (n+1)$ 为真。证毕。

25. 设 $P(n)$ 是命题：$l_0^2+l_1^2+\cdots+l_n^2=l_nl_{n+1}+2$。基础情形 $P(0)$ 和 $P(1)$ 都成立，因为 $l_0^2=2^2=2\cdot 1+2=l_0l_1+2$ 且 $l_0^2+l_1^2=2^2+1^2=1\cdot 3+2=l_1l_3+2$。现在假设 $P(n)$ 成立。则根据归纳假设，有 $l_0^2+l_1^2+\cdots+l_n^2+l_{n+1}^2=l_nl_{n+1}+2+l_{n+1}^2=l_{n+1}(l_n+l_{n+1})+2=l_{n+1}+l_{n+2}+2$。这样就证明了 $P(n+1)$ 成立。证毕。

27. 设 $P(n)$ 是命题：对整数 n 来说，这个恒等式成立。基础情形 $P(1)$ 显然为真。假设 $P(n)$ 为真，则 $\cos(n+1)x+i\sin(n+1)x=\cos(nx+x)+i\sin(nx+x)=\cos nx\cos x-\sin nx\sin x+i(\sin nx\cos x+\cos nx\sin x)=\cos x(\cos nx+i\sin nx)(\cos x+i\sin x)=(\cos x+i\sin x)^n\ (\cos x+i\sin x)=(\cos x+i\sin x)^{n+1}$。所以 $P(n+1)$ 为真，证毕。

29. a) 92　b) 91　c) 91　d) 91　e) 91　f) 91

31. 基础步骤是不正确的，因为对所显示的和来说 $n\neq 1$。

33. 设 $P(n)$ 是"若 n 个圆环中，每两个都有两个公共点，但任何三个都没有公共点，则这些圆环把平面划分成 n^2-n+2 个区域"。基础步骤：$P(1)$ 为真，因为一个圆环划分平面成 $2=1^2-1+2$ 个区域。归纳步骤：假设 $P(n)$ 为真，即具有所规定性质的 n 个圆环划分平面成 n^2-n+2 个区域。假设添加第 $n+1$ 个圆环。这个圆环与其余 n 个圆不中的每个都相交于两点，所以这些交点形成 $2n$ 段新的圆弧，每段圆弧都分割一个老的区域。因此有 $2n$ 个区域被分割，这说明比以前多出 $2n$ 个区域。因此满足所规定性质的 $n+1$ 个圆环划分平面成 $n^2-n+2+2n=(n^2+2n+1)-(n+1)+2=(n+1)^2-(n+1)+2$ 个区域。

35. 假如 $\sqrt{2}$ 是有理数。则 $\sqrt{2}=a/b$，其中 a 和 b 都是正整数。所以集合 $S=\{n\sqrt{2}\mid n\in\mathbf{N}\}\cap\mathbf{N}$ 是正整数的非空集合，因为 $b\sqrt{2}=a$ 属于 S。设 t 是 S 的最小元素，根据良序性，t 是存在的。于是对某个整数 s 来说有 $t=s\sqrt{2}$。有 $t-s=s\sqrt{2}-s=s(\sqrt{2}-1)$，所以 $t-s$

是正整数，因为$\sqrt{2}>1$。因此$t-s$属于S。矛盾，因为$t-s=s\sqrt{2}-s<s$。因此$\sqrt{2}$是无理数。

37. 假设良序性为假。设S是没有最小元素的非负整数的非空集合。设$P(n)$是命题“$i\notin S$，$i=0,1,\cdots,n$”。$P(0)$为真，因为若$0\in S$，则S有最小元素，即0。现在假设$P(n)$为真。因此$0\notin S$，$1\notin S$，…，$n\notin S$。显然$n+1$不可能属于S，因为假如它属于S，那么它就是最小元素。因此$P(n+1)$为真。根据数学归纳法原理，对所有非负整数n来说，$n\notin S$。因此$S=\varnothing$，矛盾。

39. a) 设$d=\gcd(a_1,a_2,\cdots,a_n)$。则$d$是每个$a_i$的因子，所以$d$必然是$\gcd(a_{n-1},a_n)$的因子。因此$d$是$a_1$，$a_2$，…，$a_{n-2}$和$\gcd(a_{n-1},a_n)$的公因子。为了证明$d$是这些数的最大公因子，假设$c$是它们的公因子。则对$i=12$，…，$n-2$来说，$c$是$a_i$的因子，并且$c$是$\gcd(a_{n-1},a_n)$的因子，所以它是$a_{n-1}$和$a_n$的公因子。因此$c$是$a_1$，$a_2$，…，$a_{n-1}$，$a_n$的公因子。因此$c$是$a_1$，$a_2$，…，$a_n$的最大公因子$d$的因子。所以$d$的最大公因子，与所声明的一样。

 b) 若$n=2$，则应用欧几里得算法。否则，对a_{n-1}和a_n应用欧几里得算法，得出$d=\gcd(a_{n-1},a_n)$，然后对a_1，a_2，…，a_{n-2}，d递归地应用这个算法。

41. $f(n)=n^2$。设$P(n)$是“$f(n)=n^2$”。基础步骤：$P(1)$为真，因为$f(1)=1=1^2$，这是从f的定义得出的。归纳步骤：假设$f(n)=n^2$，则$f(n+1)=f((n+1)-1)+2(n+1)-1=f(n)+2n+1=n^2+2n+1=(n+1)^2$。

43. a) λ，0，1，00，01，11，000，001，011，111，0000，0001，0011，0111，1111，00000，00001，00011，00111，01111，11111

 b) $S=\{\alpha\beta \mid \alpha$是由$m$个0所组成的串，$\beta$是由$n$个1所组成的串，$m\geq 0$，$n\geq 0\}$

45. λ，(),(()),()()

47. a) 0 b) −2 c) 2 d) 0

49.
```
procedure generate(n:非负整数)
if n 是奇数 then
  begin
    S:=S(n-1);T:=T(n-1)
  end
else if n=0 then
  begin
    S:=ø; T:={λ}
  end
else
  begin
    T₁:=T(n-2);S₁:=S(n-2)
    T:=T₁∪{(x)|x∈T₁∪S₁ 并且 l(x)=n-2}
    S:=S₁∪{xy|x∈T₁ 并且 y∈T₁∪S₁ 并且 l(xy)=n}
  end {T∪S 是长度至多为 n 的平衡串的集合}
```

51. 若起初$x\leq y$，则不执行$x:=y$，所以$x\leq y$是真的终结断言。若起初$x>y$，则执行$x:=y$，所以$x\leq y$仍是真的终结断言。

第 4 章

4.1 节

1. a) 5850 b) 343
3. a) 4^{10} b) 5^{10}
5. 42
7. 26^3
9. 676
11. 2^8
13. $n+1$（包含空串）
15. 475 255（包含空串）
17. 1 321 368 961
19. a) 128 b) 450 c) 9 d) 675
    e) 450 f) 450 g) 225 h) 75
21. a) 990 b) 500 c) 27
23. 3^{50}
25. 52 457 600
27. 20 077 200
29. a) 37 822 859 361 b) 8 204 716 800
    c) 40 159 050 880 d) 12 113 640 000
    e) 171 004 205 215 f) 321 272 407 040
    g) 6 230 721 635 h) 223 149 665
31. a) 0 b) 120 c) 720 d) 2520
33. a) 若 $n=1$，为 2；若 $n=2$，为 2；若 $n\geqslant 3$，为 0
    b) 对于 $n>1$，为 2^{n-2}；若 $n=1$，为 1；
    c) $2(n-1)$
35. $(n+1)^m$
37. 若 n 是偶数，为 $2^{n/2}$；若 n 是奇数，为 $2^{(n+1)/2}$
39. a) 240 b) 480 c) 360
41. 352
43. 147
45. 33
47. 7 104 000 000 000
49. 18
51. 17
53. 设$P(m)$是关于 m 个任务的求和法则。对于基础情形取 $m=2$。这就是 2 个任务的求和法则。现在假设$P(m)$为真，考虑 $m+1$ 个任务，T_1，T_2，…，T_{m+1}，它们可以分别用 n_1，n_2，…，n_{m+1}种方式完成，并且没有 2 个任务是同时做的。为了完成 1 个任务，

我们可以或者完成前 m 个任务中的 1 个，或者完成任务 T_{m+1}。根据 2 个任务的求和法则，完成任务的方式数是完成前 m 个任务之一的方式数与 n_{m+1} 之和。由归纳假设，这就是 $n_1+n_2+\cdots+n_m+n_{m+1}$，这正是所求的。

55. $n(n-3)/2$

4.2 节

1. 因为有 6 门课，但是只有 5 个工作日，鸽巢原理证明至少有 2 门课排在同一天。
3. a) 3　b) 14
5. 当一个整数被 4 除时只有 4 种可能的余数。给定 5 个整数，由鸽巢原理至少有 2 个整数有同样的余数。
7. 设 a，$a+1$，…，$a+n-1$ 是序列中的整数。因为 $0<(a+j)-(a+k)<n$，其中 $0\leqslant k<j\leqslant n-1$，所以整数 $(a+i) \bmod n$ 是不等的，$i=0$，1，2，…，$n-1$。由于 $(a+i) \bmod n$ 有 n 个可能的值，且在这个集合中有 n 个不同的整数，每个值恰好取 1 次。所以序列中恰好有一个整数被 n 整除。
9. 4951
11. 点 (a, b, c) 和 (d, e, f) 的连线的中点是 $((a+d)/2, (b+e)/2, (c+f)/2)$。它有整数坐标当且仅当 a 和 d 是奇偶性相同，b 和 e 的奇偶性相同。且 c 和 f 的奇偶性相同。因为奇偶性的 3 元组有 8 种可能［例如(偶,奇,偶)］，根据鸽巢原理，9 个点中至少有 2 个点有相同的奇偶性 3 元组。这 2 点连线的中点有整数坐标。
13. a) 将前 8 个正整数 2 个一组分成 4 组使得每组整数之和等于 9：{1，8}，{2，7}，{3，6} 和 {4，5}。如果从前 8 个整数中选 5 个整数，由鸽巢原理至少有 2 个整数取自同一组。这 2 个整数之和等于 9。

    b) 不正确。例如，取 {1，2，3，4}。
15. 21 251
17. 设 d_j 是 $jx-N(jx)$，其中 $N(jx)$ 是距 jx 最近的整数，$1\leqslant j\leqslant n$。每个 d_j 是在 $-1/2$ 和 $1/2$ 之间的无理数。先设 n 是偶数，n 是奇数的情况更凌乱一些。考虑 n 个区间 $\{x \mid j/n<x<(j+1)/n\}$，$\{x \mid -(j+1)/n<x<-j/n\}$，$j=0$，1，…，$(n/2)-1$。如果对于某个 j，d_j 属于区间 $\{x \mid 0<x<1/n\}$ 或区间 $\{x \mid -1/n<x<0\}$，命题得证。如果不是，则因为有 $n-2$ 个区间和 n 个数 d_j，由鸽巢原理存在一个区间 $\{x \mid (k-1)/n<x<k/n\}$ 包含 d_r 和 d_s，其中 $r<s$。只要证明 $(s-r)x$ 距最近的整数在 $1/n$ 之内这个证明就完成了。
19. 4，3，2，1，8，7，6，5，12，11，10，9，16，15，14，13
21. **Procedure** *long* $(a_1, \cdots, a_n$：正整数)

```
{首先找最长的递增子序列}
max := 0; set := 00…00 {n 位}
for i := 0 to 2^n
begin
   last := 0; count := 0; OK := true
   for j := 1 to n
```

```
        begin
          if set (j) = 1 then
            begin
            if a_j > last then last := a_j
            count := count + 1
          end
          else OK := false
        end
      if count > max then
        begin
          max := count
          best := set
        end
      set := set + 1 {二进制加}
    end {max是长度并且 best 给出这个序列}
```

{对于递减子序列重复进行，唯一不同的是用 $a_j < last$ 代替 $aj > last$，且用 $last := \infty$ 代替 $last := 0$}

23．根据对称性只需证明第一个语句。设 A 是其中的一个人。A 在其他 9 个人中或至少有 4 个朋友，或至少有 6 个敌人（因为 $3+5<9$）。假设是第 1 种情况，B，C，D 和 E 是 A 的朋友。如果这些人中任有 2 个人是朋友，我们已经找到 3 个人彼此都是朋友。否则 $\{B, C, D, E\}$ 是彼此都是敌人的 4 人集合。在第 2 种情况下，设 $\{B, C, D, E, F, G\}$ 是 A 的敌人的集合。由例 11，在 B，C，D，E，F 和 G 中或有 3 个人彼此都是朋友，或有 3 个人彼此都是敌人，这样，他们和 A 构成 4 人的彼此都是敌人的集合。

25．姓的前 3 个字母和生日有 6 432 816 种可能性，因此，由鸽巢原理存在至少 $\lceil 25\ 000\ 000/6\ 432\ 816 \rceil = 4$ 个人，他们姓的前 3 个字母和生日都相同。

27．18

29．因为有 6 台计算机，连接到同一台计算机的其他机器数在 0～5 之间，包含 0 和 5 在内。但是，0 和 5 不能同时出现。为此只需注意到下面的事实：如果某台计算机不与其他机器连接，那么没有 1 台计算机连接到所有其他的 5 台机器；如果某台计算机连接到所有其他的 5 台机器，那么就没有不与其他机器相连的计算机。于是 1 台计算机连接的机器数至多有 5 种可能，由鸽巢原理在 6 台计算机中至少有 2 台连接的其他机器数相等。

31．设 a_i 是到第 i 小时为止所完成的比赛数，那么 $1 \leqslant a_1 < a_2 < \cdots < a_{75} \leqslant 125$，进而有 $25 \leqslant a_1 + 24 < a_2 + 24 < \cdots < a_{75} + 24 \leqslant 149$。有 150 个数 a_1，a_2，…，a_{75}，a_1+24，a_2+24，…，$a_{75}+24$，由鸽巢原理至少有 2 个数相等。由于所有的 a_i 是不等的，所有的 a_i+24 也是不等的，因此对于某个 $i>j$ 有 $a_i = a_j + 24$。于是从第 $j+1$ 到第 i 小时恰有 24 场比赛。

33．使用推广的鸽巢原理，对于 $s \in S$，把 $|S|$ 个物体 $f(s)$ 放到 $|T|$ 个盒子里，T 中的每个元素一个盒子。

35．a）如果在这个班里一年级学生少于 9 个，二年级学生少于 9 个，三年级学生也少于 9

个，则每个年级的学生都不超过 8 个，那么学生总数至多 24 个，这与班里有 25 个学生矛盾。

b) 如果一年级学生少于 3 个，二年级学生少于 19 个，三年级学生少于 5 个，那么至多 2 个一年级学生，至多 18 个二年级学生，至多 4 个三年级学生，学生总数至多是 24，与班上有 25 个学生矛盾。

37. a) 假设对所有的 k，$i_k \leq n$，那么由推广的鸽巢原理，数 i_1，i_2，…，i_{n^2+1} 中至少有 $\lceil (n^2+1)/n \rceil = n+1$ 个数相等。

b) 如果 $a_{k_j} < a_{k_{j+1}}$，那么 a_{k_j} 和以 $a_{k_{j+1}}$ 开始长度为 $i_{k_{j+1}}$ 的递增子序列组成的子序列与 $i_{k_j} = i_{k_{j+1}}$ 矛盾。于是，$a_{k_j} > a_{k_{j+1}}$。

c) 如果没有长度大于 n 的递增子序列，那么使用 a)和 b)，得到长度为 $n+1$ 的递减子序列 $a_{k_1} > a_{k_2} > \cdots > a_{k_{n+1}}$。

4.3 节

1. abc，acb，bac，bca，cab，cba
3. 720
5. a) 120　b) 720　c) 8
   d) 6720　e) 40 320　f) 3 628 800
7. 15 120
9. 1320
11. $2(n!)^2$
13. 65 780
15. $2^{100} - 5051$
17. a) 94 109 400　b) 941 094
   c) 3 764 376　d) 90 345 024
   e) 114 072　f) 2328
   g) 24　h) 79 727 040
   i) 3 764 376　j) 109 440
19. a) 12 650　b) 303 600
21. a) 37 927　b) 18 915
23. a) 122 523 030　b) 72 930 375
   c) 223 149 655　d) 100 626 625
25. 54 600
27. 45
29. 912
31. 11 232 000
33. $C(n+1,k) = \frac{(n+1)!}{k!(n+1-k)!}$

$$= \frac{(n+1)}{k}\frac{n!}{(k-1)!(n-(k-1))!}$$

$= (n+1)C(n,k-1)/k$

这个恒等式与$C(n,0)=1$一起给出了一个递归定义。

35. $x^5+5x^4y+10x^3y^2+10x^2y^3+5xy^4+y^5$

37. 101

39. $-2^{10}C(19,9)=-94\ 595\ 072$

41. $-2^{101}3^{99}C(200,99)$

43. 若 $k\equiv 2 \pmod 3$且$-100\leqslant k\leqslant 200$，为$(-1)^{(200-k)/3}C(100,(200-k)/3)$；否则为 0。

45. 1 9 36 84 126 126 84 36 9 1

47. $C(n,k-1)+C(n,k)$

$$= \frac{n!}{(k-1)!(n-k+1)!}+\frac{n!}{k!(n-k)!}$$

$$= \frac{n!}{k!(n-k+1)!}\cdot[k+(n-k+1)]$$

$$= \frac{(n+1)!}{k!(n+1-k)!}=C(n+1,k)$$

49. a) $C(n+r+1,r)$通过选择 0 的位置计数了选择含 r 个 0 和 $n+1$ 个 1 的序列的方式数。另一方面，假设第 $j+1$ 项是等于 1 的最后一项，因此 $n\leqslant j\leqslant n+r$。一旦确定了最后一个 1 的位置，我们就可以确定在最后一个 1 前边的 j 个位置中有哪些位置放 0，在这个范围内有 n 个 1 和 $j-n$ 个 0。根据求和法则有 $\sum_{j=n}^{n+r}C(j,j-n)=\sum_{k=0}^{r}C(n+k,k)$ 种方式做这件事。

b) 设$p(r)$是需证明的命题。基础步骤是等式$C(n,0)=C(n+1,0)$，即 $1=1$。假设$P(r)$为真，那么$\sum_{k=0}^{r+1}C(n+k,k)=\sum_{k=0}^{r}C(n+k,k)+C(n+r+1,r+1)=C(n+r+1,r)+C(n+r+1,r+1)=C(n+r+2,r+1)$，由归纳假设和帕斯卡等式得证。

51. 首先以 n 种方式选择领导，然后以 2^{n-1}种方式选择委员会的其他成员，于是有$n2^{n-1}$种方式选择委员会及其领导。另一方面，选择一个 k 人委员会的方式数是$C(n,k)$。一旦我们选择了 k 个人的委员会，则存在 k 种方式选择它的领导，于是有 $\sum_{k=1}^{n}kC(n,k)$ 种方式选择这个委员会及其领导。从而有$\sum_{k=1}^{n}kC(n,k)=n2^{n-1}$。

53. 设这个集合有 n 个元素。由定理 7 有$C(n,0)-C(n,1)+C(n,2)-\cdots+(-1)^nC(n,n)=0$，从而得到$C(n,0)+C(n,2)+C(n,4)+\cdots=C(n,1)+C(n,3)+C(n,5)+\cdots$。左边给出了具有偶数个元素的子集个数，且右边给出了具有奇数个元素的子集个数。

55. a) 一条所求的路径由 m 次向右的移动和 n 次向上的移动构成。每一条这种路径可以由一个 m 个 0 和 n 个 1 的 $m+n$ 位二进制串表示，其中 0 表示一次向右的移动，1 表示一次向上的移动。

b) 恰包含 n 个 1 的 $m+n$ 位二进制串个数等于$C(m+n,n)=C(m+n,m)$，因为可以通过指定 n 个 1 的位置或指定 m 个 0 的位置确定这样的串。

57. 由练习 55 所描述的 n 步长的路径条数等于 2^n，就是 n 位二进制串的个数。另一方面，

练习 55 中所描述的 n 步长的路径一定在坐标之和等于 n 的某个点结束，比如说 $(n-k,k)$ 点，其中 k 在 0 和 n 之间，包含 0 和 n 在内。由练习 55，这种在 $(n-k,k)$ 点结束的路径数等于 $C(n-k+k,k)=C(n,k)$。于是 $\sum_{k=0}^{n}C(n,k)=2^n$。

59. 练习 55 所描述的从 (0,0) 到 $(n+1,r)$ 的路径数等于 $C(n+r+1,r)$，但是这种路径由向上走 j 步开始，其中 j 满足 $0\leqslant j\leqslant r$。由向上走 j 步开始的这种路径数等于练习 55 所描述的从 $(1,j)$ 到 $(n+1,r)$ 的路径数，而这又与从 (0,0) 到 $(n,r-j)$ 的路径数一样，根据练习 55 它等于 $C(n+r-j,r-j)$。因为 $\sum_{j=0}^{r}C(n+r-j,r-j)=\sum_{k=0}^{r}C(n+k,k)$，从而 $\sum_{k=0}^{r}C(n+k,k)=C(n+r+1,r)$。

61. 543

63. a) $C(n+1,2)$　　b) $C(n+2,3)$
    c) $C(2n-2,n-1)$　　d) $C(n-1,\lfloor(n-1)/2\rfloor)$
    e) 在帕斯卡三角的第 n 行中的最大奇数项
    f) $C(3n-3,n-1)$

4.4 节

1. 1/13
3. 1/2
5. 1/2
7. 1/64
9. 47/52
11. $1/C(52,5)$
13. $1-(C(48,5)/C(52,5))$
15. $C(13,2)C(4,2)C(4,2)C(44,1)/C(52,5)$
17. $10\ 240/C(52,5)$
19. $1\ 302\ 540/C(52,5)$
21. 1/64
23. 8/25
25. a) $1/C(50,6)=1/15\ 890\ 700$
    b) $1/C(52,6)=1/20\ 358\ 520$
    c) $1/C(56,6)=1/32\ 468\ 436$
    d) $1/C(60,6)=1/50\ 063\ 860$
27. a) 139128/319865　　b) 212667/511313
    c) 151340/386529　　d) 163647/446276
29. $1/C(100,8)$
31. a) 9/19　　b) 81/361　　c) 1/19
    d) 1 889 568/2 476 099　　e) 48/361
33. 掷 3 个骰子。
35. 这个竞争者选的门是在不知道奖在何处的情况下随机选的，但是由主持人选的门不是随

机选的，因为他总是避免打开有奖的门。这就使得基于对称性的论据是无效的。

4.5 节

1. $p(T)=1/4$，$p(H)=3/4$
3. $p(1)=p(3)=p(5)=p(6)=1/16$；$p(2)=p(4)=3/8$
5. 9/49
7. 由于$p(E\cup F)=p(E)+p(F)-p(E\cap F)$，且$p(E\cup F)\leqslant 1$，从而得到$1\geqslant p(E)+p(F)-p(E\cap F)$。由这个不等式可以断言$p(E)+p(F)\leqslant 1+p(E\cap F)$。
9. 我们将使用数学归纳法证明这个不等式对 $n\geqslant 2$ 成立。设$p(n)$是语句 $p(\bigcup_{j=1}^{n}E_j)\leqslant\sum_{j=1}^{n}p(E_j)$。$P(2)$为真。因为$p(E_1\cup E_2)=p(E_1)+p(E_2)-p(E_1\cap E_2)\leqslant p(E_1)+p(E_2)$。现在假设$p(n)$为真，使用基础情形和归纳假设得$p(\bigcup_{j=1}^{n+1}E_j)\leqslant p(\bigcup_{j=1}^{n}E_j)+p(E_{n+1})\leqslant\sum_{j=1}^{n+1}p(E_j)$，这就证明了$p(n+1)$为真。根据数学归纳法命题得证。
11. 因为 $E\cup\overline{E}$ 是整个样板空间S，事件 F 可以被分成两个不交的事件：$F=S\cap F=(E\cup\overline{E})\cap F=(E\cap F)\cup(\overline{E}\cap F)$，这里使用了分配律。由于这两个事件是不交的，故$p(F)=p((E\cap F)\cup(\overline{E}\cap F))=p(E\cap F)+p(\overline{E}\cap F)$。使用事实$p(E\cap F)=p(E)$。$p(F)$(已知 E 和F 是独立的)，在两边都减去$p(E\cap F)$并且分解因式得$p(F)(1-p(E))=p(\overline{E}\cap F)$。由于$1-p(E)=p(\overline{E})$，这就得到$p(\overline{E}\cap F)=p(\overline{E})\cdot p(F)$，正如所求。
13. a) $1-365/366\cdot 364/366\cdots(367-n)/366$
    b) 23
15. 1/4
17. 3/8
19. a) 不独立　b) 不独立　c) 不独立
21. 3/16
23. a) $1/32=0.03125$　b) $0.49^5\approx 0.02825$
    c) 0.03795012
25. a) 5/8　b) 0.627649　c) 0.6431
27. a) p^n　b) $1-p^n$
    c) $p^n+np^{n-1}(1-p)$
    d) $1-[p^n+np^{n-1}(1-p)]$
29. 5/3
31. 336/49
33. 170
35. $(4n+6)/3$
37. pq^{n-1}　$[q=1-p]$
39. $1-(1-p)^n$
41. 1
43. 5/2
45. a) 0　b) 1

47. $E(X)/a = \sum_r (r/a) \cdot p(X = r) \geqslant \sum_{r \geqslant a} 1 \cdot p(X = r) = p(X \geqslant a)$

49. a) 10/11　　b) 0.9984

51. $V(X+Y) = E((X+Y)^2) - E(X+Y)^2 = E(X^2 + 2XY + Y^2) - (E(X) + E(Y))^2$
$= E(X^2) + 2E(XY) + E(Y^2) - E(X)^2 - 2E(X)E(Y) - E(Y)^2$
$= E(X^2) - E(X^2) + 2(E(XY) - E(X)E(Y)) + E(Y^2) - E(Y)^2$
$= V(X) + 2\mathrm{Cov}(X, Y) + V(Y)$

4.6 节

1. 243
3. 26^6
5. 125
7. 35
9. a) 1716　　b) 50 388　　c) 2 629 575
   d) 330　　9 724
11. 9
13. 4 504 501
15. a) 10 626　　b) 1 365　　c) 11 649　　d) 106
17. 2 520
19. 302 702 400
21. 30 492
23. $C(59, 50)$
25. 35
27. 83 160
29. 63
31. 19 635
33. 210
35. 27 720
37. $52!/(7!^5 17!)$
39. $24 \cdot 13^4/(52 \cdot 51 \cdot 50 \cdot 49)$
41. a) $C(k+n-1, n)$　　b) $(k+n-1)!/(k-1)!$
43. 对于第 1 个盒子有 $C(n, n_1)$ 种方式选择 n_1 个物体。一旦选了这些物体以后，对于第 2 个盒子有 $C(n-n_1, n_2)$ 种方式选择物体。类似地，对于第 3 个盒子有 $C(n-n_1-n_2, n_3)$ 种方式选择物体。按照这种方式继续下去直到在 $C(n-n_1-n_2-\cdots-n_{k-1}, n_k) = C(n_k, ln_k) = 1$ 种方式选择最后一个盒子的物体(由于 $n_1 + n_2 + \cdots + n_k = n$)。由乘积法则构成全部分配的方式数是 $C(n, n_1)C(n-n_1, n_2)C(n-n_1-n_2, n_3)\cdots C(n-n_1-n_2-\cdots-n_{k-1}, n_k)$，它等于 $n!/(n_1!\ n_2!\ \cdots n_k!)$，正如直接化简所显示的。
45. a) 由于 $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_r$，从而 $x_1 + 0 < x_2 + 1 < \cdots < x_r + r - 1$。这个不等式是严格的，因为只要 $x_j \leqslant x_{j+1}$，就有 $x_j + j - 1 < x_{j+1} + j$。因为 $1 \leqslant x_j \leqslant n + r - 1$，这个序列是由 T 中选择 r 个不同的元素构造出来的。

b) 假设 $1 \leqslant x_1 < x_2 < \cdots < x_r \leqslant n + r - 1$。令 $y_k = x_k - (k-1)$。那么不难看出对于 $k = 1, 2, \cdots, r-1$ 有 $y_k < y_{k+1}$，且对于 $k = 1, 2, \cdots, r$ 有 $1 \leqslant y_k \leqslant n$。从而 $\{y_1, y_2, \cdots, y_k\}$ 是 S 的允许重复的 r-组合。

c) 从 a)和 b) 得出在 S 的允许重复的 r-组合与具有 $n + r - 1$ 个元素的集合 T 的 r-组合之间存在一一对应。可以断言存在着 $C(n + r - 1, r)$ 个 S 的允许重复的 r-组合。

47. 5

49. 在展开式中的项形如 $x_1^{n_1} x_2^{n_2} \cdots x_m^{n_m}$，其中 $n_1 + n_2 + \cdots + n_m = n$。产生这样的项是因为从 n_1 个因式中选择了 x_1，从 n_2 个因式中选择了 x_2，…从 n_m 个因式中选择了 x_m。这可以用 $C(n; n_1, n_2, \cdots, n_m)$ 种方式做到，因为一种选择就是 n_1 个标签“1”，n_2 个标签“2”…和 n_m 个标签“m”的排列。

51. 2520

4.7 节

1. a) 2134　　b) 54 132
   c) 12 534　　d) 45 312
   e) 6 714 253　　f) 31 542 678

3. 1234，1243，1324，1342，1423，1432，2134，2143，2314，2341，2413，2431，3124，3142，3214，3241，3412，3421，4123，4132，4213，4231，4312，4321

5. {1，2，3}，{1，2，4}，{1，2，5}，{1，3，4}，{1，3，5}，{1，4，5}，{2，3，4}，{2，3，5}，{2，4，5}，{3，4，5}

7. 表示下一个最大的 r-组合的二进制串一定与表示原来的 r-组合的二进制串在第 i 位不同，因为第 $i+1, \cdots, r$ 位被最大可能的数占据。此外，如果我们想要一个比原来的组合大的组合，$a_i + 1$ 是可以放在第 i 位的最小可能的数。那么 $a_i + 2, \cdots, a_i + r - i + 1$ 是从第 $i+1$ 到 r 位所允许的最小的数。于是我们生成了下一个 r 组合。

9. 123，132，213，231，312，321，124，142，214，241，412，421，125，152，215，251，512，521，134，143，314，341，413，431，135，153，315，351，513，531，145，154，415，451，514，541，234，243，324，342，423，432，235，253，325，352，523，532，245，254，425，452，524，542，345，354，435，453，534，543

11. 我们将通过证明它有一个逆来证明这是一个双射。给定一个小于 $n!$ 的正整数，设 $a_1, a_2, \cdots, a_{n-1}$ 是它的康托儿数字。把 n 放在位置 $n - a_{n-1}$，因此很清楚 a_{n-1} 是在排列中跟在 n 后面且小于 n 的整数个数。然后把 $n-1$ 放在空位置 $(n-1) - a_{n-2}$，这里我们已经标号了空位置 $1, 2, \cdots, n-1$（除去 n 已经在的位置）。继续下去直到 1 放在留下的唯一的空位置。因为我们已经构造了一个逆，这个对应是一个双射。

13. **Procedure** *Cantor Permutation* (n, i：整数满足 $n \geqslant 1$ 和 $0 \leqslant i < n!$)

```
x := n
for j := 1 to n
    p_j := 0
for k := 1 to n - 1
    begin
```

```
c := ⌊x/(n − k)!⌋; x := x − c(n − k)!; h := n
while p_h ≠ 0
    h := h − 1
for j := 1 to c
begin
    h := h − 1
    while p_h ≠ 0
        h := h − 1
end
p_h := n − k + 1
end
h := 1
while p_h ≠ 0
    h := h + 1
p_h := 1
{p_1 p_2 ··· p_n 是对应于 i 的排列}
```

补充练习

1. a) 151 200　b) 1 000 000　c) 210　d) 5005
2. 3^{100}
5. 24 600
7. a) 4 060　b) 2688　c) 25 009 600
9. a) 192　b) 301　c) 300　d) 300
11. 639
13. 最大可能的和是 240，且最小可能的和是 15。所以可能的和的个数是 226。因为一个 10 元集存在 252 个 5 元子集，由鸽巢原理至少两个有同样的和。
15. a) 50　b) 50　c) 14　d) 5
17. 设 a_1，a_2，…，a_m 是整数，且令 $d_i=\sum_{j=1}^{i}a_j$。如果对于某个 i，$d_i \equiv 0 \pmod m$，命题得证。否则 $d_1 \bmod m$，$d_2 \bmod m$，…，$d_m \bmod m$ 是 m 个在 $\{1, 2, \cdots, m-1\}$ 中的整数。由鸽巢原理，对于某个 $1\leqslant k<1\leqslant m$ 有 $d_k=d_1$，于是 $\sum_{j=k+1}^{1}a_j=d_1-d_k\equiv 0 \pmod m$
19. 有理数 a/b 的十进制展开式可以通过 b 除 a 得到，其中 a 写成含有小数点的十进制小数，且后面跟随着无数个 0。基础步骤是找出商的下一个十进制数字，即$\lfloor r/b \rfloor$，其中 r 是余数和从被除数移下来的下一位。当前的余数由前面的余数减去商的前面数字的 b 倍得到。最后被除数没有数字时移下 0。此外，由于只存在 b 个可能的余数。由鸽巢原理，在某一步将得到与前面某一步相同的结果。从这一步向后计算一定遵循着相同的模式。特别是商将被重复。
21. a) 125 970　b) 20　c) 141 120 525

d) 141 120 505 e) 177 100 f) 141 078 021

23. $4^{13}/C(52,13)$

25. a) 10 b) 8 c) 7

27. 确定选择 n 元集的哪些 r 个元素的方法数和确定不选哪些 $n-r$ 个元素的方法数一样。

29. $C(n+2,r+1)=C(n+1,r+1)+C(n+1,r)=2C(n+1,r+1)-C(n+1,r+1)$
$+C(n+1,r)=2C(n+1,r+1)-(C(n,r+1)+C(n,r))+(C(n,r)+C(n,r-1))$
$=2C(n+1,r+1)-C(n,r+1)+C(n,r-1)$

31. 由二项式定理，$3^n=(2+1)^n=\sum_{k=0}^{n}C(n,k)1^{n-k}2^k=\sum_{k=0}^{n}C(n,k)2^k$

33. $C(n+1,5)$

35. a) $1/C(52,13)$ b) $4/C(52,13)$
c) $2\,944\,656/C(52,13)$ d) $35\,335\,872/C(52,13)$
e) $1\,244\,117\,160/C(52,13)$ f) $29\,858\,811\,840/C(52,13)$

37. $\dfrac{(m-1)(n-1)+\gcd(m,n)-1}{mn-1}$

39. a) $p(E_1\cap E_2)=p(E_1)p(E_2)$, $p(E_1\cap E_3)=p(E_1)p(E_3)$, $p(E_2\cap E_3)$
$=p(E_2)p(E_3)$, $p(E_1\cap E_2\cap E_3)=p(E_1)p(E_2)p(E_3)$
b) 是 c) 是 d) 2^n-n-1

41. 事件 $E\cap F_i$，$i=1,2,\cdots,n$ 是互相排斥的并且覆盖了 E 出现的所有条件。因此，$p(E)=\sum_{i=1}^{n}p(E\cap F_i)$。由条件概率的定义知道 $p(E\cap F_i)=p(E|F_i)p(F_i)$，并且 $p(E|F_i)=p(E\cap F_i)/p(E)$。因此

$$(p(F_j|E)=\frac{p(E\cap F_j)}{p(E)}=\frac{p(E|F_j)p(F_j)}{\sum_{i=1}^{n}p(E\cap F_i)}=\frac{p(E|F_j)p(F_j)}{\sum_{i=1}^{n}p(E|F_i)p(F_i)}$$

43. $V(aX+b)=E((aX+b)^2)-E(aX+b)^2=E(a^2X^2+2abX+b^2)-(aE(X)+b)^2$
$=E(a^2X^2)+E(2abX)+E(b^2)-(a^2E(X)^2+2abE(X)+b^2)=a^2E(X^2)+2abE(X)$
$+b^2-a^2E(X)^2-2abE(X)-b^2=a^2(E(X^2)-E(X)^2)=a^2V(X)$

45. 3 491 888 400

47. 5^{24}

49. a) 45 b) 57 c) 12

51. a) 386 b) 56 c) 512

53. 如果 $n<m$，为 0；如果 $n\geq m$，为 $C(n-1,n-m)$

55. **Procedure** *next Permutation* (n: 正整数，a_1，a_2，$\cdots$，a_r: 不超过 n 的正整数，且 $a_1a_2\cdots a_r\neq nn\cdots n$)

```
i := r
while a_i = n
  begin
    a_i := 1
    i := i - 1
```

end
$a_i := a_i + 1$
{$a_1 a_2 \cdots a_r$ 是按照字典次序的下一个排列}

第 5 章

5.1 节

1. a) 2, 12, 72, 432, 2592
   b) 2, 4, 16, 256, 65, 536
   c) 1, 2, 5, 11, 26
   d) 1, 1, 6, 27, 204
   e) 1, 2, 0, 1, 3
3. a) 是 b) 不是 c) 不是 d) 是 e) 是 f) 是 g) 不是 h) 不是
5. a) $a_n = 2\cdot 3^n$ b) $a_n = 2n+3$ c) $a_n = 1 + n(n+1)/2$ d) $a_n = n^2+4n+4$
   e) $a_n = 1$ f) $a_n = (3^{n+1}-1)/2$ g) $a_n = 5n!$ h) $a_n = 2^n n!$
7. a) $a_n = 3a_{n-1}$ b) 5 904 900
9. a) $a_n = n + a_{n-1}$, $a_0 = 0$ b) $a_{12} = 78$ c) $a_n = n(n+1)/2$
11. 设$P(n)$是 "$H_n = 2n-1$," 基础步骤：$P(1)$为真，因为 $H_1 = 1$。归纳步骤：假设 $H_n = 2^n - 1$，那么因为 $H_{n+1} = 2H_n + 1$，从而得到$H_{n+1} = 2(2^n-1)+1 = 2^{n+1}-1$。
13. a) $a_n = 2a_{n-1} + a_{n-5}$, $n \geqslant 5$
    b) $a_0 = 1$, $a_1 = 2$, $a_2 = 4$, $a_3 = 8$, $a_4 = 16$
    c) 1217
15. 9494
17. a) $a_n = a_{n-1} + a_{n-2} + 2^{n-2}$, $n \geqslant 2$
    b) $a_0 = 0$, $a_1 = 0$ c) 94
19. a) $a_n = a_{n-1} + a_{n-2} + a_{n-3}$, $n \geqslant 3$
    b) $a_0 = 1$, $a_1 = 2$, $a_2 = 4$ c) 81
21. a) $a_n = a_{n-1} + a_{n-2}$, $n \geqslant 2$
    b) $a_0 = 1$, $a_1 = 1$ c) 34
23. a) $a_n = 2a_{n-1} + 2a_{n-2}$, $n \geqslant 2$
    b) $a_0 = 1$, $a_1 = 3$ c) 448
25. a) $a_n = 2a_{n-1} + a_{n-2}$, $n \geqslant 2$
    b) $a_0 = 1$, $a_1 = 3$ c) 239
27. a) $a_n = 2a_{n-1}$, $n \geqslant 2$
    b) $a_1 = 3$ c) 96
29. a) $a_n = a_{n-1} + a_{n-2}$, $n \geqslant 2$
    b) $a_0 = 1$, $a_1 = 1$ c) 89
31. a) $R_n = n + R_{n-1}$, $R_0 = 1$

b) $R_n = n(n+1)/2+1$

33. a) $S_n = S_{n-1} + (n^2 - n + 2)/2$，$S_0 = 1$

b) $S_n = (n^3 + 5n + 6)/6$

35. 64

37. a) $a_n = 2a_{n-1} + 2a_{n-2}$　b) $a_0 = 1$，$a_1 = 3$　c) 1224

39. 很清楚对于 $m \geqslant 1$ 有 $S(m,1) = 1$。如果 $m \geqslant n$，那么一个从 m 元集到 n 元集的非映上函数可以由值域的大小和值域中的元素来确定。这里值域的大小是在 $1 \sim n-1$ 之间的整数 k，包含1 和 $n-1$ 在内；而值域中的元素可以用 $C(n,k)$ 种方式选取，并且构成到这个值域的映上函数可以有 $S(m,k)$ 种方式。因此存在 $\sum_{k=1}^{n-1} C(n,k)S(m,k)$ 个非映上函数。因总共存在 n^m 个函数，所以 $S(m,n) = n^m - \sum_{k=1}^{n-1} C(n,k)S(m,k)$。

41. a) $C_5 = C_0C_4 + C_1C_3 + C_2C_2 + C_3C_1 + C_4C_0 = 1\cdot 14 + 1\cdot 5 + 2\cdot 2 + 5\cdot 1 + 14\cdot 1 = 42$

b) $C(10,5)/6 = 42$

43. $J(1)=1, J(2)=1, J(3)=3, J(4)=1, J(5)=3, J(6)=5, J(7)=7, J(8)=1, J(9)=3, J(10)=5, J(11)=7, J(12)=9, J(13)=11, J(14)=13, J(15)=15, J(16)=1$

45. 首先，假设人数是偶数，比如说是 $2n$。在转子一圈并且回到第 1 个人 y 以后，由于在偶数位置的人已经被排除，则恰好有 n 个人留下来并且现在在位置 i 的人就是初始在位置 $2i-1$ 的人。因此，生还者[初始在位置 $J(2n)$]现在的位置是 $J(n)$，这就是在位置 $2J(n)-1$ 的人。所以，$J(2n) = 2J(n) - 1$。类似地，当有奇数个人时，比如说 $2n+1$ 个人，那么在转子一圈以后排除第 1 个人，有 n 个人留下来并且现在在位置 i 的人就是以前在位置 $2i+1$ 的人。因此，生还者将是现在占据位置 $J(n)$ 的人，即初始在位置 $2J(n)+1$ 的人。所以，$J(2n+1) = 2J(n) + 1$。基础情形是 $J(1) = 1$。

47. 73 977 3617

49. 下面的 9 次移动求解了这个难题：从柱 1 到柱 2 移动盘 1；从柱 1 到柱 3 移动盘 2；从柱 2 到柱 3 移动盘 1；从柱 1 到柱 2 移动盘 3；从柱 1 到柱 4 移动盘 4；从柱 2 到柱 4 移动盘 3；从柱 3 到柱 2 移动盘 1；从柱 3 到柱 4 移动盘 2；从柱 2 到柱 4 移动盘 1。要明白至少需要 9 次移动，首先注意到不管有多少根柱子至少 7 次移动是需要的：3 次用来拿走盘子，1 次用来移动最大的盘 4，并且还要 3 次移动来放好盘子。另外至少还需要 2 次移动，因为要从柱 1 到柱 4 移动盘 4，其他 3 个盘子一定在柱 2 和柱 3，所以至少需要 1 次移动来放好它们，再 1 次移动来拿走它们。

51. 基础情形是显然的。如果 $n>1$，算法由 3 步构成。第 1 步，由归纳假设，使用 $R(n-k)$ 次移动把最小的 $n-k$ 个盘子移到柱 2。然后使用通常的 3 个柱子的汉诺塔算法，用 2^k-1 次移动把剩下的盘子(最大的 k 个盘子)移到柱 4，移动时不使用柱 2。然后再由归纳法，用 $R(n-k)$ 次移动把最小的 $n-k$ 个盘子移到柱 4。对于这种移动所有的柱子都可以用，因为最大的盘子在柱 4，对移动没有妨碍。这就建立了递推关系。

53. 首先观察到 $R(n) = \sum_{j=1}^{n}(R(j) - R(j-1))$ (因为化简这个和并且 $R(0)=0$)。由练习 52，这就是关于 j 的值域对 $2^{k'-1}$ 的求和，因此这个和是 $\sum_{i=1}^{k} i2^{i-1}$。如果 n 不是一个三角形数，那么当 $i=k$ 时最后某些值丢失，并且这就是在给定表达式中最后的项所表示的。

55. 由练习 53，$R(n)$不大于 $\sum_{i=1}^{k} i2^{i-1}$。可以证明这个和等于$(k+1)2^k - 2^{k+1} + 1$，因此它不比$(k+1)2^k$ 大。因为$n > k(k-1)/2$，可以用二次公式证明对于所有的$n > 1$ 有 $k < 1 + \sqrt{2n}$。因而对于所有的 $n > 2$，$R(n)$以$(1+\sqrt{2n}+1)2^{1+\sqrt{2n}} < 8\sqrt{n}2^{\sqrt{2n}}$为上界。所以$R(n)$是$O(\sqrt{n}2^{\sqrt{2n}})$。

57. 由练习 34 知$a_n = 2^{n-1}$,$\nabla^2 a_n = \nabla a_n - \nabla a_{n-1} = 2^{n-2} - 2^{n-3} = 2^{n-3}$

59. $a_n - 2\nabla a_n + \nabla^2 a_n$

$$= a_n - 2(a_n - a_{n-1}) + (\nabla a_n - \nabla a_{n-1})$$
$$= -a_n + 2a_{n-1} + ((a_n - a_{n-1}) - (a_{n-1} - a_{n-2}))$$
$$= -a_n + 2a_{n-1} + (a_n - 2a_{n-1} + a_{n-2})$$
$$= a_{n-2}$$

61. $a_n = a_{n-1} + a_{n-2}$

$$= (a_n - \nabla a_n) + (a_n - 2\nabla a_n + \nabla^2 a_n)$$
$$= 2a_n - 3\nabla a_n + \nabla^2 a_n,$$

或　$a_n = 3\nabla a_n - \nabla^2 a_n$,

5.2 节

1. a) 3 阶　b) 不是　c) 4 阶　d) 不是　e) 不是　f) 2 阶　g) 不是
3. a) $a_n = 3\cdot 2^n$　b) $a_n = 2$
   c) $a_n = 3\cdot 2^n - 2\cdot 3^n$　d) $a_n = 6\cdot 2^n - 2\cdot n2^n$
   e) $a_n = n(-2)^{n-1}$　f) $a_n = 2^n - (-2)^n$
   g) $a_n = (1/2)^{n+1} - (-1/2)^{n+1}$
5. $a_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}$
7. $(2^{n+1} + (-1)^n)/3$
9. a) $P_n = 1.2P_{n-1} + 0.45P_{n-2}$，$P_0 = 100\ 000$，$P_1 = 120\ 000$
   b) $P_n = (250\ 000/3)(3/2)^n + (5\ 000\ 013)(-3/10)^n$
11. a) 基础步骤：对于 $n=1$ 有 $1=0+1$，且对于 $n=2$ 有 $3=1+2$。归纳步骤：假设对于 $k \leqslant n$ 为真，那么$L_{n+1} = L_n + L_{n-1} = f_{n-1} + f_{n+1} + f_{n-2} + f_n = (f_{n-1} + f_{n-2}) + (f_{n+1} + f_n) = f_n + f_{n+2}$。
    b) $L_n = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n$
13. $a_n = 8(-1)^n - 3(-2)^n + 4.3^n$
15. $a_n = 5 + 3(-2)^n - 3^n$
17. 设$a_n = C(n,0) + C(n-1,1) + \cdots + C(n-k,k)$，其中$k = \lfloor n/2 \rfloor$。首先，假设 n 是偶数，使得 $k = n/2$，且最后项是$C(k,k)$，由帕斯卡恒等式有$a_n = 1 + C(n-2,0) + C(n-2,1) + C(n-3,1) + C(n-3,2) + \cdots + C(n-k,k-2) + C(n-k,k-1) + 1 = 1 + C(n-2,1) + C(n-3,2) + \cdots + C(n-k,k-1) + C(n-2,0) + C(n-3,1) + \cdots +$

$C(n-k,k-2)+1=a_{n-1}+a_{n-2}$，因为$\lfloor(n-1)/2\rfloor=k-1=\lfloor(n-2)/2\rfloor$。当 n 是奇数时有类似的计算。因此，对于所有的正整数 n，$n\geqslant 2$，$\{a_n\}$满足递推关系 $a_n=a_{n-1}+a_{n-2}$。此外，$a_1=C(1,0)=1$ 和 $a_2=C(2,0)+C(1,1)=2$，这就是f_2 和 f_3。从而得到对于所有的正整数 n，$a_n=f_{n+1}$。

19. $a_n=(n^2+3n+5)(-1)^n$

21. $(a_{1,0}+a_{1,1}n+a_{1,2}n^2+a_{1,3}n^3)+(a_{2,0}+a_{2,1}n+a_{2,2}n^2)(-2)^n+(a_{3,0}+a_{3,1}n)3^n+a_{4,0}(-4)^n$

23. a) $3a_{n-1}+2^n=3(-2^n)+2^n=2^n(-3+1)=-2^{n+1}=a_n$
 b) $a_n=\alpha 3^n-2^{n+1}$
 c) $a_n=3^{n+1}-2^{n+1}$

25. a) $A=-1$，$B=-7$
 b) $a_n=\alpha 2^n-n-7$
 c) $a_n=11\cdot 2^n-n-7$

27. a) $p_2n^2+p_1n+p_0$
 b) $n^2p_0(-2)^n$
 c) $n^2(p_1n+p_0)2^n$
 d) $(p_2n^2+p_1n+p_0)4^n$
 e) $n^2(p_2n^2+p_1n+p_0)(-2)^n$
 f) $n^2(p_4n^4+p_3n^3+p_2n^2+p_1n+p_0)2^n$
 g) p_0

29. a) $a_n=\alpha 2^n+3^{n+1}$
 b) $a_n=-2\cdot 2^n+3^{n+1}$

31. $a_n=\alpha 2^n+\beta 3^n-n\cdot 2^{n+1}+3n/2+21/4$

33. $a_n=(\alpha+\beta n+n^2+n^3/6)2^n$

35. $a_n=-4\cdot 2^n-n^2/4-5n/2+1/8+(39/8)3^n$

37. $a_n=n(n+1)(n+2)/6$

39. a) 1，-1，i，$-i$
 b) $a_n=\frac{1}{4}-\frac{1}{4}(-1)^n+\frac{2+i}{4}i^n+\frac{2-i}{4}(-i)^n$

41. a) 使用关于 f_n 的公式可看出

$$\left|f_n-\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n\right|<1/\sqrt{5}<1/2$$

 这意味着 f_n 是最接近$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$ 的整数。
 b) 当 n 是偶数时较小；当 n 是奇数时较大。

43. $a_n=f_{n-1}+2f_n-1$

45. a) $a_n=3a_{n-1}+4a_{n-2}$，$a_0=2$，$a_1=6$
 b) $a_n=(4^{n+1}+(-1)^n)/5$

47. a) $a_n=2a_{n-1}+(n-1)10000$
 b) $a_n=70000 2^{n-1}-10000n-10000$

49. $a_n=5n^2/12+13n/12+1$

51. 见参考文献［Ma93］第 11 章第 5 节。

5.3 节

1. 14

3. 第 1 步是$(1110)_2(1010)_2 = (2^4+2^2)(11)_2(10)_2 + 2^2((11)_2-(10)_2)((10)_2-(10)_2) + (2^2+1)(10)_2 \cdot (10)_2$。这个积是$(10001100)_2$。
5. $C = 50\ 665C + 729 = 33\ 979$
7. a) 2 b) 4 c) 7
9. a) 79 b) 48 829 c) 30 517 579
11. $O(\log n)$
13. $O(n\log_3 2)$
15. 5
17. 由于 $k=\log_b n$，从而有 $f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j c(n/b^j)^d = a^k f(1) + \sum_{j=0}^{k-1} cn^d = a^k f(1) + kcn^d = a^{\log_b n} f(1) + c(\log_b n)n^d = n^{\log_b a} f(1) + cn^d \log_b n = n^d f(1) + cn^d \log_b n$。
19. 设 $k=\log_b n$，其中 n 是 b 的幂。基础步骤：如果 $n=1$ 且 $k=0$，那么 $c_1 n^d + c_2 n\log_b a = c_1 + c_2 = = b^d c/(b^d - a) + f(1) + b^d c/(a - b^d) = f(1)$。归纳步骤：假设对于 k 为真，其中 $n = b_k$。那么对于 $n = b^{k+1}$，$f(n) = af(n/b) + cn^d = a\{[b^d c/(b^d - a)](n/b)^d + [f(1) + b^d c/(a - b^d)] \cdot (n/b)\log_b a\} + cn^d = b^d c/(b^d - a)n^d a/b^d + [f(1) + b^d c/(a - b^d)]n\log_b a + cn^d = n^d[ac/(b^d - a) + c(b^d - a)/(b^d - a)] + [f(1) + b^d c/(a - b^d)] \cdot n\log_b a = [(b^d c)/(b^d - a)]n^d + [f(1) + b^d c/(a - b^d)]n\log_b a$
21. 如果 $a > b^d$，那么 $\log_b a > d$，所以第二项为主，给出$O(n\log_b a)$。
23. $O(n\log_4 5)$
25. $O(n^3)$

5.4 节

1. $f(x) = 2(x^6 - 1)/(x - 1)$
3. a) $f(x) = 2x(1 - x^6)/(1 - x)$
   b) $x^3/(1 - x)$
   c) $x(1 - x^3)$
   d) $2/(1 - 2x)$
   e) $(1 + x)^7$
   f) $2/(1 + x)$
   g) $(1/(1 - x)) - x^2$
   h) $x^3/(1 - x)^2$
5. a) $5/(1 - x)$
   b) $1/(1 - 3x)$
   c) $2x^3/(1 - x)$
   d) $(3 - x)/(1 - x)^2$
   e) $(1 + x)^8$
   f) $1/(1-x)^5$
7. a) $a_0 = -64$，$a_1 = 144$，$a_2 = -108$，$a_3 = 27$，对于所有的 $n \geq 4$ 有 $a_n = 0$

b) 非零系数只有 $a_0=1$，$a_3=3$，$a_6=3$，$a_9=1$

c) $a_n=5^n$

d) 对于 $n\geqslant 3$ 有 $a_n=(-3)^{n-3}$，$a_0=a_1=a_2=0$

e) $a_0=8$，$a_1=3$，$a_2=2$，对于比 2 大的 n，n 为奇数有 $a_n=0$，n 为偶数有 $a_n=1$

f) 如果 n 是 4 的正整数倍，$a_n=1$；如果 $n<4$ 有 $a_n=-1$；否则 $a_n=0$

g) 对于 $n\geqslant 2$ 有 $a_n=n-1$，$a_0=a_1=0$

h) $a_n=2^{n+1}/n!$

9. a) 6 b) 3 c) 9 d) 0 e) 5

11. a) 1024 b) 11 c) 66 d) 292 864 e) 20 412

13. 10

15. 50

17. 20

19. $f(x)=1/((1-x)(1-x^2)(1-x^5)(1-x^{10}))$

21. 15

23. a) $x^4(1+x+x^2+x^3)^2/(1-x)$

b) 6

25. a) 在 $1/((1-x^3)(1-x^4)(1-x^{20}))$ 的幂级数展开式中 x^r 的系数

b) $1/(1-x^3-x^4-x^{20})$

c) 7

d) 3224

27. a) 3 b) 29 c) 29 d) 242

29. a) 10 b) 49 c) 2 d) 4

31. a) $G(x)-a_0-a_1x-a_2x^2$ b) $G(x^2)$

c) $x^4G(x)$ d) $G(2x)$

e) $\left(\int_0^x G(t)dt\right)/x$ f) $G(x)/(1-x)$

33. $a_k=2\cdot 3^k-1$

35. $a_k=18\cdot 3^k-12\cdot 2^k$

37. $a_k=k^2+8k+20+(6k-18)2^k$

39. 设 $G(x)=\sum_{k=0}^{\infty}f_kx^k$。在对和式的序标移位并将序列求和以后，易得 $G(x)-xG(x)-x^2G(x)=f_0+(f_1-f_0)x+\sum_{k=2}^{\infty}(f_k-f_{k-1}-f_{k-2})x^k=0+x+\sum_{k=2}^{\infty}0x^k$，由此得 $G(x)-xG(x)-x^2G(x)=x$。求解 $G(x)$ 得 $G(x)=x/(1-x-x^2)$。由部分分式的方法证明 $x/(1-x-x^2)=(1/\sqrt{5})[1/(1-\alpha x)-1/(1-\beta x)]$，其中 $\alpha=(1+\sqrt{5})/2$ 且 $\beta=(1-\sqrt{5})/2$。由事实 $1/(1-\alpha x)=\sum_{k=0}^{\infty}\alpha^kx^k$，得 $G(x)=(1/\sqrt{5})\cdot\sum_{k=0}^{\infty}(\alpha^k-\beta^k)x^k$，因此 $f_k=(1/\sqrt{5})(\alpha^k-\beta^k)$。

41. a) 设 $G(x)=\sum_{n=0}^{\infty}C_nx^n$ 是 $\{C_n\}$ 的生成函数。那么 $G(x)^2=\sum_{n=0}^{\infty}(\sum_{k=0}^{n}C_kC_{n-k})x^n=\sum_{n=1}^{\infty}(\sum_{k=0}^{n-1}C_kC_{n-1-k})x^{n-1}=\sum_{n=1}^{\infty}C_nx^{n-1}$。因此 $xG(x)^2=\sum_{n=1}^{\infty}C_nx^n$，这就推

出$xG(x)^2-G(x)+1=0$。应用二次方程求根公式得$G(x)=\frac{1\pm\sqrt{1-4x}}{2x}$，在这个公式中选择减号，因为选择加号会导致除以零。

b) 由练习40 知$(1-4x)^{-1/2}=\sum_{n=0}^{\infty}\binom{2n}{n}x^n$，逐项积分（根据微积分的定理是允许的）得$\int_0^x(1-4t)^{-1/2}dt=\sum_{n=0}^{\infty}\frac{1}{n+1}\binom{2n}{n}x^{n+1}=x\sum_{n=0}^{\infty}\frac{1}{n+1}\binom{2n}{n}x^n$。因为$\int_0^x(1-4t)^{-1/2}dt=\frac{1-\sqrt{1-4x}}{2}=xG(x)$，由系数相等得$C_n=\frac{1}{n+1}\binom{2n}{n}$。

43. 对等式$(1+x)^{m+n}=(1+x)^m(1+x)^n$应用二项式定理，证明$\sum_{r=0}^{m+n}C(m+n,r)x^r=\sum_{r=0}^{m}C(m,r)x^r\cdot\sum_{r=0}^{n}C(n,r)x^r=\sum_{r=0}^{m+n}\left(\sum_{k=0}^{r}C(m,r-k)C(n,k)\right)x^r$。比较系数就得到所需要的恒等式。

45. a) $2e^x$　b) e^{-x}　c) e^{3x}　d) xe^x+e^x　e) $(e^x-1)/x$

47. a) $a_n=(-1)^n$　b) $a_n=3\cdot 2^n$

c) $a_n=3^n-3\cdot 2^n$　d) $a_n=(-2)^n,n\geqslant 2;a_1=-3,a_0=2$

e) $a_n=(-2)^n+n!$　f) $a_n=(-3)^n+n!\cdot 2^n,n\geqslant 2^n;a_0=1,a_1=-2$

g) $a_n=0$，n 是奇数；$a_n=n!/(n/2)!$，n 是偶数

49. a) $a_n=6a_{n-1}+8^{n-1}$，$n\geqslant 1$；$a_0=1$

b) 相关的线性齐次递推关系的通解是 $a_n^{(h)}=\alpha 6^n$，特解是 $a_n^{(p)}=\frac{1}{2}8^n$。因此通解是 $a_n=\alpha 6^n+\frac{1}{2}8^n$。使用初始条件得 $\alpha=\frac{1}{2}$，因此$a_n=(6^n+8^n)/2$

c) 设$G(x)=\sum_{k=0}^{\infty}a_kx^k$。使用关于$\{a_k\}$的递推关系可证明$G(x)-6xG(x)=(1-7x)/(1-8x)$，因此$G(x)=(1-7x)/((1-6x)(1-8x))$。由部分分式得$G(x)=(1/2)/(1-6x)+(1/2)(1-8x)$。利用表 1 得$a_n=(6^n+8^n)/2$

51. $\frac{1}{1-x}\cdot\frac{1}{1-x^2}\cdot\frac{1}{1-x^3}\cdots$

53. $(1+x)(1+x^2)(1+x^3)\cdots$

55. 因为$(1+x)(1+x^2)(1+x^3)\cdots=\frac{1-x^2}{1-x}\cdot\frac{1-x^4}{1-x^2}\cdot\frac{1-x^6}{1-x^3}\cdots=\frac{1}{1-x}\frac{1}{1-x^3}\frac{1}{1-x^5}\cdots$，练习 52 和练习 53 的生成函数相等。

57. a) $G_X(1)=\sum_{k=0}^{\infty}p(X=k)1^k=\sum_{k=0}^{\infty}p(X=k)=1$

b) $G'_X(1)=\frac{d}{dx}\sum_{k=0}^{\infty}p(X=k)\cdot x^k\Big|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot k\cdot x^{k-1}\Big|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot k=E(X)$

c) $G''_X(1)=\frac{d^2}{dx^2}\sum_{k=0}^{\infty}p(X=k)\cdot x^k\Big|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot k(k-1)\cdot x^{k-2}\Big|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot(k^2-k)=V(X)+E(X)^2-E(X)$。与 b) 组合就得到所需的结果

59. a) $G(x)p^m/(1-qx)^m$
    b) $V(x)=mq/p^2$

5.5 节

1. a) 30 b) 29 c) 24 d) 18
3. 1%
5. a) 300 b) 150 c) 175 d) 100
7. 492
9. 974
11. 55
13. 248
15. 50 138
17. 234
19. $|A_1\cup A_2\cup A_3\cup A_4\cup A_5|=|A_1|+|A_2|+|A_3|+|A_4|+|A_5|-|A_1\cap A_2|-|A_1\cap A_3|-|A_1\cap A_4|-|A_1\cap A_5|-|A_2\cap A_3|-|A_2\cap A_4|-|A_2\cap A_5|-|A_3\cap A_4|-|A_3\cap A_5|-|A_4\cap A_5|+|A_1\cap A_2\cap A_3|+|A_1\cap A_2\cap A_4|+|A_1\cap A_2\cap A_5|+|A_1\cap A_3\cap A_4|+|A_1\cap A_3\cap A_5|+|A_1\cap A_4\cap A_5|+|A_2\cap A_3\cap A_4|+|A_2\cap A_3\cap A_5|+|A_2\cap A_4\cap A_5|+|A_3\cap A_4\cap A_5|-|A_1\cap A_2\cap A_3\cap A_4|-|A_1\cap A_2\cap A_3\cap A_5|-|A_1\cap A_2\cap A_4\cap A_5|-|A_1\cap A_3\cap A_4\cap A_5|-|A_2\cap A_3\cap A_4\cap A_5|+|A_1\cap A_2\cap A_3\cap A_4\cap A_5|$
21. $|A_1\cup A_2\cup A_3\cup A_4\cup A_5\cup A_6|=|A_1|+|A_2|+|A_3|+|A_4|+|A_5|+|A_6|-|A_1\cap A_2|-|A_1\cap A_3|-|A_1\cap A_4|-|A_1\cap A_5|-|A_1\cap A_6|-|A_2\cap A_3|-|A_2\cap A_4|-|A_2\cap A_5|-|A_2\cap A_6|-|A_3\cap A_4|-|A_3\cap A_5|-|A_3\cap A_6|-|A_4\cap A_5|-|A_4\cap A_6|-|A_5\cap A_6|$
23. $p(E_1\cup E_2\cup E_3)=p(E_1)+p(E_2)+p(E_3)-p(E_1\cap E_2)-p(E_1\cap E_3)-p(E_2\cap E_3)+p(E_1\cap E_2\cap E_3)$
25. 4972/71 295
27. $p(E_1\cup E_2\cup E_3\cup E_4\cup E_5)=p(E_1)+p(E_2)+p(E_3)+p(E_4)+p(E_5)-p(E_1\cap E_2)-p(E_1\cap E_3)-p(E_1\cap E_4)-p(E_1\cap E_5)-p(E_2\cap E_3)-p(E_2\cap E_4)-p(E_2\cap E_5)-p(E_3\cap E_4)-p(E_3\cap E_5)-p(E_4\cap E_5)+p(E_1\cap E_2\cap E_3)+p(E_1\cap E_2\cap E_4)+p(E_1\cap E_2\cap E_5)+p(E_1\cap E_3\cap E_4)+p(E_1\cap E_3\cap E_5)+p(E_1\cap E_4\cap E_5)+p(E_2\cap E_3\cap E_4)+p(E_2\cap E_3\cap E_5)+p(E_2\cap E_4\cap E_5)+p(E_3\cap E_4\cap E_5)$
29. $p\left(\bigcup_{i=1}^{n}E_i\right)=\sum_{1\leqslant i\leqslant n}p(E_i)-\sum_{1\leqslant i<j\leqslant n}p(E_i\cap E_j)+\sum_{1\leqslant i<j<k\leqslant n}p(E_i\cap E_j\cap E_k)-\cdots+(-1)^{n+1}p\left(\bigcap_{i=1}^{n}E_i\right)$

5.6 节

1. 75
3. 6

5. 46

7. 9875

9. 540

11. 2100

13. 1854

15. a) $D_{100}/100!$　b) $100D_{99}/100!$　c) $C(100,2)/100!$
　d) 0　e) $1/100!$

17. 2 170 680

19. 由练习 18 有$D_n - nD_{n-1} = -(D_{n-1} - (n-1)D_{n-2})$。通过迭代以及$D_2 = 1, D_1 = 0$，得

$$D_n - nD_{n-1} = -(D_{n-1} - (n-1)D_{n-2}) = -(-(D_{n-2} - (n-2)D_{n-3}))$$
$$= D_{n-2} - (n-2)D_{n-3} = \cdots = (-1)^n(D_2 - 2D_1) = (-1)^n$$

21. n 是奇数。

23. $\phi(n) = n - \sum_{i=1}^{m} \frac{n}{p_i} + \sum_{1 \leq i < j \leq m} \frac{n}{p_i p_j} - \cdots \pm \frac{n}{p_1 p_2 \cdots p_m} = n \prod_{i=1}^{m} \left(a - \frac{1}{p_i}\right)$

25. 4

27. 从 m 元集到 n 元集存在 n^m 个函数，从 m 元集到 n 元集有$C(n,1)(n-1)^m$ 个函数恰好缺少 1 个元素，$C(n,2)(n-2)^m$ 个函数恰好缺少 2 个元素，继续下去，有$C(n,n-1)1^m$ 个函数恰好缺少 $n-1$ 个元素。因此由容斥原理存在$n^m - C(n,1)(n-1)^m + C(n,2)(n-2)^m - \cdots + (-1)^{n-1}C(n,n-1)1^m$ 个映上函数。

补充练习

1. a) $A_n = 4A_{n-1}$　b) $A_1 = 40$　c) $A_n = 10 \cdot 4^n$

3. a) $M_n = M_{n-1} + 160\ 000$
　b) $M_1 = 186\ 000$
　c) $M_n = 160\ 000n + 26\ 000$
　d) $T_n = T_{n-1} + 160\ 000n + 26\ 000$
　e) $T_n = 80\ 000n^2 + 106\ 000n$

5. a) $a_n = a_{n-2} + a_{n-3}$
　b) $a_1 = 0$，$a_2 = 1$，$a_3 = 1$
　c) $a_{12} = 12$

7. a) 2　b) 5　c) 8　d) 16

9. $a_n = 2^n$

11. $a_n = 2 + 4n/3 + n^2/2 + n^3/6$

13. $a_n = a_{n-2} + a_{n-3}$

15. $O(n^4)$

17. $O(n)$

19. a) $18n + 18$　b) 18　c) 0

21. $\Delta(a_n b_n) = a_{n+1}b_{n+1} - a_n b_n = a_{n+1}(b_{n+1} - b_n) + b_n(a_{n+1} - a_n) = a_{n+1}\Delta b_n + b_n \Delta a_n$

23. a) 设$G(x)=\sum_{n=0}^{\infty}a_nx^n$，那么$G'(x)=\sum_{n=1}^{\infty}na_nx^{n-1}=\sum_{n=0}^{\infty}(n+1)a_{n+1}x^n$。因此，$G'(x)-G(x)=\sum_{n=0}^{\infty}((n+1)a_{n+1}-a_n)x^n=\sum_{n=0}^{\infty}x^n/n!=e^x$，正如所求。易见$G(0)=a_0=1$。

b) 我们有$(e^{-x}G(x))'=e^{-x}G'(x)-e^{-x}G(x)=e^{-x}(G'(x)-G(x))=e^{-x}e^x=1$。因此，$e^{-x}G(x)=x+c$，其中$c$是常数。从而$G(x)=xe^x+ce^x$。由于$G(0)=1$，所以$c=1$。

c) 我们有$G(x)=\sum_{n=0}^{\infty}(x^{n+1}/n!)+\sum_{n=0}^{\infty}(x^n/n!)=\sum_{n=1}^{\infty}(x^n/(n-1)!)+\sum_{n=0}^{\infty}(x^n/n!)$。由此，对于所有的$n\geqslant1$，$a_n=1/(n-1)!+1/n!$，且$a_0=1$。

25. 7

27. 110

29. 0

31. a) 19 b) 65 c) 122 d) 167 e) 168

33. $D_{n-1}/(n-1)!$

35. 11/32

第 6 章

6.1 节

1. a) $\{(0,0),(1,1),(2,2),(3,3)\}$
   b) $\{(1,3),(2,2),(3,1),(4,0)\}$
   c) $\{(1,0),(2,0),(2,1),(3,0),(3,1),(3,2),(4,0),(4,1),(4,2),(4,3)\}$
   d) $\{(1,0),(1,1),(1,2),(1,3),(2,0),(2,2),(3,0),(3,3),(4,0)\}$（假设 0 不除以 0）
   e) $\{(0,1),(1,0),(1,1),(1,2),(1,3),(2,1),(2,3),(3,1),(3,2),(4,1),(4,3)\}$
   f) $\{(1,2),(2,1),(2,2)\}$
3. a) 传递的
   b) 自反的，对称的，传递的
   c) 对称的
   d) 反对称的
   e) 自反的，对称的，反对称的，传递的
   f) 没有这些性质
5. a) 对称的
   b) 对称的，传递的
   c) 对称的
   d) 自反的，对称的，传递的
   e) 自反的，传递的
   f) 自反的，对称的，传递的
   g) 反对称的
   h) 反对称的，传递的

7. c)，d)，f)

9. 是，例如，在{1，2}上的关系{(1，1)}

11. a)

13. 2^{mn}

15. a) {(a，b)|b 整除 a}

　b) {(a，b)|a 不整除 b}

17. f^{-1}的图

19. a) {(a，b)|要求 a 读书 b 或者 a 已经读过书 b}

　b) {(a，b)|要求 a 读书 b 并且 a 已经读过书 b}

　c) {(a，b)|或者要求 a 读书 b 但是 a 还没有读过书 b，或者不要求 a 读书 b 但是 a 已经读过书 b}

　d) {(a，b)|要求 a 读书 b 但是 a 还没有读过书 b}

　e) {(a，b)|不要求 a 读书 b 但是 a 已经读过书 b}

21. $S \circ R$ = {(a,b)|a 是 b 的父母且 b 有一个兄弟姐妹}

　$R \circ S$ = {(a,b)|a 是 b 的叔伯或阿姨}

23. 8

25. a) $2^{n(n+1)/2}$　b) $2^n 3^{n(n-1)/2}$

　c) $3^{n(n-1)/2}$　d) $2^{n(n-1)}$

　e) $2^{n(n-1)/2}$　f) $2^{n^2} - 2 \cdot 2^{n(n-1)}$

27. 可能没有这样的 b。

29. 若 R 是对称的且$(a,b) \in R$，那么$(b,a) \in R$，因此$(a,b) \in R^{-1}$，从而有$R \subseteq R^{-1}$。类似地有$R^{-1} \subseteq R$，于是$R = R^{-1}$。相反，若$R = R^{-1}$且$(a,b) \in R$，那么$(a,b) \in R^{-1}$，从而$(b,a) \in R$，于是 R 是对称的。

31. R 是自反的当且仅当对所有的$a \in A$有$(a,a) \in R$，当且仅当对所有的$a \in A$有$(a,a) \in R^{-1}$[因为$(a,a) \in R$当且仅当$(a,a) \in R^{-1}$]，从而当且仅当R^{-1}是自反的。

33. 使用数学归纳法。对于 $n=1$ 结果是平凡的。假设 R 是自反和传递的。由定理 1，$R^{n+1} \subseteq R$。为得到 $R \subseteq R^{n+1} = R^n \circ R$，令$(a,b) \in R$。由归纳假设，$R^n = R$ 并且因此是自反的。于是$(b,b) \in R^n$，从而$(a,b) \in R^{n+1}$。

35. 使用数学归纳法。$n=1$ 时结果是平凡的。假设 R^n 是自反的，那么对所有的 $a \in A$ 有$(a,a) \in R^n$ 且$(a,a) \in R$。于是对所有的 $a \in A$ 有$(a,a) \in R^n \circ R = R^{n+1}$。

37. 不是，例如，$R = \{(1,2),(2,1)\}$。

6.2 节

1. {(1,2,3),(1,2,4),(1,3,4),(2,3,4)}

3. (Nadir，122，34，底特律，08:10)，(Acme，221，22，丹佛，08:17)，(Acme，122，33，安克雷奇，08:22)，(Acme，323，34，檀香山，08:30)，(Nadir，199，13，底特律，08:47)，(Acme，222，22，丹佛，09:10)，(Nadir，322，34，底特律，09:44)

5. 航线与航班号，航线与起飞时间

7. $P_{3,5,6}$

9.

航线	目的地
Nadir	底特律
Acme	丹佛
Acme	安克雷奇
Acme	檀香山

11.

供货商	零件号	项目	数量	颜色代码
23	1092	1	2	2
23	1101	3	1	1
23	9048	4	12	2
31	4975	3	6	2
31	3477	2	25	2
32	6984	4	10	1
32	9191	2	80	4
33	1001	1	14	8

6.3 节

1.

a) $\begin{bmatrix}1&1&1\\0&0&0\\0&0&0\end{bmatrix}$ b) $\begin{bmatrix}0&1&0\\1&1&0\\0&0&1\end{bmatrix}$ c) $\begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}$ d) $\begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix}$

3. R 是反自反的当且仅当每个对角线元素都是 0。

5. 把每个 0 变成 1 并且把每个 1 变成 0。

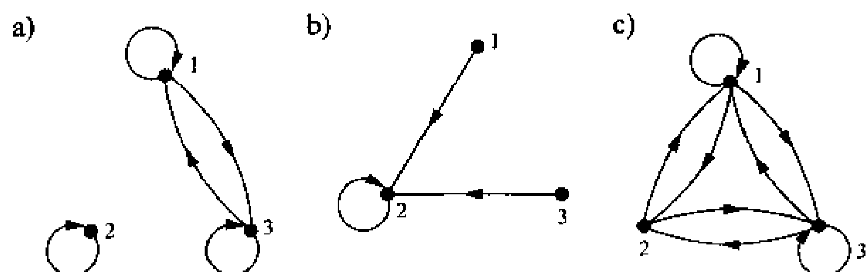
7.

a) $\begin{bmatrix}0&1&1\\1&1&0\\1&0&1\end{bmatrix}$ b) $\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$ c) $\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

9.

a) $\begin{bmatrix}0&0&1\\1&1&0\\0&1&1\end{bmatrix}$ b) $\begin{bmatrix}1&1&0\\0&1&1\\1&1&1\end{bmatrix}$ c) $\begin{bmatrix}0&1&1\\1&1&1\\1&1&1\end{bmatrix}$

11.



13. {(a,b),(a,c),(b,c),(c,b)}

15. {(*a*,*a*),(*a*,*b*),(*a*,*c*),(*b*,*a*),(*b*,*b*),(*b*,*c*),(*c*,*a*),(*c*,*b*),(*d*,*d*)}

17. a) 只是反自反的

    b) 只是自反的

    c) 只是对称的

19. 用数学归纳法证明。$n=1$ 是平凡的。假设对 n 为真。因为 $R^{n+1}=R^n\circ R$，它的矩阵是 $\mathbf{M}_R\odot\mathbf{M}_{R^n}$。由归纳假设就是 $\mathbf{M}_R\odot\mathbf{M}_R^{[n]}=\mathbf{M}_R^{[n+1]}$。

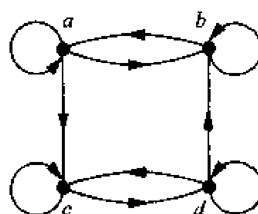
6.4 节

1. a) {(0,0),(0,1),(1,1),(1,2),(2,0),(2,2),(3,0),(3,3)}

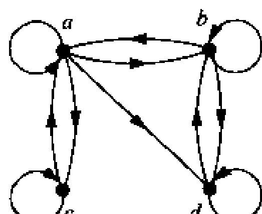
   b) {(0,1),(0,2),(0,3),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2),(3,0)}

3. {(a,b)|*a* 整除 *b* 或 *b* 整除 *a*}

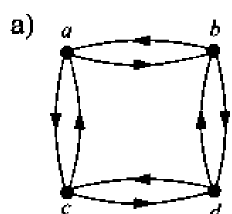
5.

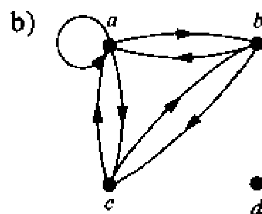


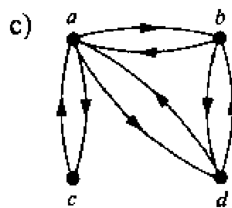
7.



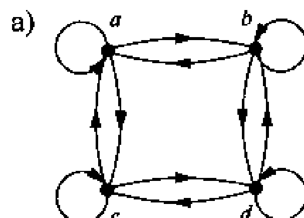
9.

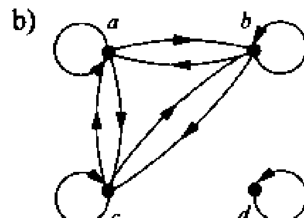
a) 

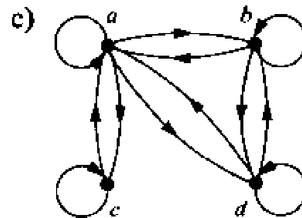
b) 

c) 

11.

a) 

b) 

c) 

13. R 的对称闭包是 $R\cup R^{-1}$。$\mathbf{M}_{R\cup R^{-1}}=\mathbf{M}_R\vee\mathbf{M}_{R^{-1}}=\mathbf{M}_R\vee\mathbf{M}_R^t$。

15. 仅当 R 是反自反关系时，它就是它自己的闭包。

17. a, a, a, a; b, c, c, b; c, b, c, c; c, c, b, c; c, c, c, c; d, e, e, d; e, d, e, e; e, e, d, e; e, e, e, e

19. a) {(1,1),(1,5),(2,3),(3,1),(3,2),(3,3),(3,4),(4,1),(4,5),(5,3),(5,4)}

    b) {(1,1),(1,2),(1,3),(1,4),(2,1),(2,5),(3,1),(3,3),(3,4),(3,5),(4,1),(4,2),

(4,3),(4,4),(5,1),(5,3),(5,5)}

c) {(1,1),(1,3),(1,4),(1,5),(2,1),(2,2),(2,3),(2,4),(3,1),(3,2),(3,3),(3,4),(3,5),(4,1),(4,3),(4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)}

d) {(1,1),(1,2),(1,3),(1,4),(1,5),(2,1),(2,3),(2,4),(2,5),(3,1),(3,2),(3,3),(3,4),(3,5),(4,1),(4,2),(4,3),(4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)}

e) {(1,1),(1,2),(1,3),(1,4),(1,5),(2,1),(2,2),(2,3),(2,4),(2,5),(3,1),(3,2),(3,3),(3,4),(3,5),(4,1),(4,2),(4,3),(4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)}

f) {(1,1),(1,2),(1,3),(1,4),(1,5),(2,1),(2,2),(2,3),(2,4),(2,5),(3,1),(3,2),(3,3),(3,4),(3,5),(4,1),(4,2),(4,3),(4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)}

21. a) 如果有学生 c，使得 a 和 c 有一门公共课程，b 和 c 也有一门公共课程。

b) 如果有两个学生 c 和 d，使得 a 和 c 有一门公共课程，c 和 d 有一门公共课程，d 和 b 也有一门公共课程。

c) 如果存在学生序列 s_0，s_1，…，s_n，$n \geq 1$，使得 $s_0 = a$，$s_n = b$，且对于每个 $i = 1, 2, \cdots, n$，s_i 与 s_{i-1} 有一门公共课程。

23. 从 $(R^*)^{-1} = (\bigcup_{n=1}^{\infty} R^n)^{-1} = \bigcup_{n=1}^{\infty} (R^n)^{-1} = \bigcup_{n=1}^{\infty} R^n = R$ 得证。

25.

a) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ b) $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ c) $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ d) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$

27. 答案与练习 25 一样。

29. a) {(1,1),(1,2),(1,4),(2,2),(3,3),(4,1),(4,2),(4,4)}

b) {(1,1),(1,2),(1,4),(2,1),(2,2),(2,4),(3,3),(4,1),(4,2),(4,4)}

c) {(1,1),(1,2),(1,4),(2,1),(2,2),(2,4),(3,3),(4,1),(4,2),(4,4)}

31. 算法 1：$O(n^{3.8})$；算法 2：$O(n^3)$

33. 初始令 $\mathbf{A} := \mathbf{M}_R \vee \mathbf{I}_n$ 且只做 **for** i **to** $n-1$ 循环。

35. a) 因为 R 是自反的，每个包含它的关系也一定是自反的。

b) {(0,0),(0,1),(0,2),(1,1),(2,2)}和{(0,0),(0,1),(1,0),(1,1),(2,2)}都包含 R 且都有奇数个元素，但是其中的每一个都不是另一个的子集。

6.5 节

1. a) 等价关系

b) 不是自反的，也不是传递的

c) 等价关系

d) 不是传递的

e) 不是对称的，也不是传递的

3. a) 等价关系

   b) 不是传递的

   c) 不是自反的，不是对称的，也不是传递的

   d) 等价关系

   e) 不是自反的，也不是传递的

5. a) 因为$f(x)=f(x)$，$(x,x)\in R$。所以 R 是自反的。$(x,y)\in R$ 当且仅当$f(x)=f(y)$，当且仅当$f(y)=f(x)$，当且仅当$(y,x)\in R$。因此 R 是对称的。如果$(x,y)\in R$ 和$(y,z)\in R$，那么$f(x)=f(y)$和$f(y)=f(z)$，因此$f(x)=f(z)$。于是$(x,z)\in R$，这就证明 R 是传递的。

   b) 对于 f 的值域中的b 得到的集合$f^{-1}(b)$。

7. 设 x 是长度至少为 3 的串，由于 x 与自己在前 3 位相同，$(x,x)\in R$，因此 R 是自反的。假设$(x,y)\in R$，那么 x 与 y 在前 3 位相同，因此 y 和 x 也在前 3 位相同，于是$(y,x)\in R$。如果(x,y)和(y,z)在 R 中，那么 x 和 y 在前 3 位相同，y 和 z 也在前 3 位相同。因此 x 和 z 在前 3 位相同，从而$(x,z)\in R$。于是 R 是传递的。

9. 命题 p 等价于 q 意味着 p 和 q 在它们的真值表中有相同的项。R 是自反的，因为 p 与 p 的真值表相同。R 是对称的，因为若 p 与 q 有相同的真值表，那么 q 与 p 也有相同的真值表。如果 p 与 q 在它们的真值表中有相同的项并且 q 与 r 在它们的真值表中有相同的项，那么 p 与 r 在它们的真值表中也有相同的项，因此 R 是传递的。

11. 不是

13. 不是

15. R 是自反的，因为二进制串 s 和它自己有相同数量的 1。R 是对称的，因为 s 和 t 有相同数量的 1，则 t 与 s 也有相同数量的 1。R 是传递的，因为如果 s 与 t 有相同数量的 1，t 与 u 有相同数量的 1，那么 s 与 u 也有相同数量的 1。

17. a) 相同年龄的人的集合

    b) 具有相同父母的人的集合

19. 所有恰好具有 2 个 1 的二进制串的集合

21. a) $\{s \mid s$ 是 3 位二进制串$\}$

    b) $\{s1 \mid s$ 是 3 位二进制串$\}$

    c) $\{s11 \mid s$ 是 3 位二进制串$\}$

    d) $\{s1010 \mid s$ 是 3 位二进制串$\}$

23. $\{6n+k \mid n\in \mathbf{Z}\}$，$k\in\{0,1,2,3,4,5\}$

25. a) 不是　b) 是　c) 是　d) 不是

27. $[0]_6\subseteq[0]_3$，$[1]_6\subseteq[1]_3$，$[2]_6\subseteq[2]_3$，$[3]_6\subseteq[0]_3$，$[4]_6\subseteq[1]_3$，$[5]_6\subseteq[2]_3$

29. $\{(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c),(d,d),(d,e),(e,d),(e,e)\}$

31. a) $\mathbf{Z}$　b) $\{n+\frac{1}{2} \mid n\in\mathbf{Z}\}$

33. a) R 是自反的，因为任何涂色方案旋转 360 度还是原来的方案。由于每个旋转是两个翻转的合成，反之，两个翻转的合成是一个旋转。由这个事实可以说明 R 是对称和

传递的。(C_1,C_2)属于 R 当且仅当 C_2 可以通过翻转的合成由 C_1 得到。如果(C_1,C_2)属于 R，那么由于翻转合成的逆也是翻转的合成（按照相反的次序），(C_2,C_1)也属于 R。于是 R 是对称的。假设(C_1,C_2)和(C_2,C_3)都属于 R，取每种情况下翻转的合成得到的仍是翻转的合成，固(C_1,C_3)属于 R，R 是传递的。

b) 我们用长度为 4 的序列表示涂色，r 和 b 分别代表红和蓝。我们按照左上方格、右上方格、左下方格、右下方格的次序列出表示颜色的字母。等价类是：{rrrr}，{bbbb}，(rrrb, rrbr, rbrr, brrr)，{bbbr, bbrb, brbb, rbbb}，{rbbr, brrb, rrbb, bbrr}，{brbr, rbrb}。

35. 5

37. 是

39. R

41. 首先构成 R 的自反闭包，然后构成这个自反闭包的对称闭包，最后构成自反闭包的对称闭包的传递闭包。

43. $p(0)=1, p(1)=1, p(2)=2, p(3)=5, P(4)=15, p(5)=52, p(6)=203, p(7)=877, p(8)=4140, p(9)=21147, p(10)=115975$

6.6 节

1. a) 是　b) 不是　c) 是　d) 不是

3. 不是

5. 是

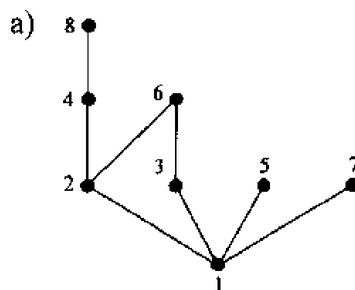
7. a) $\{(0,0),(1,0),(1,1),(2,0),(2,1),(2,2)\}$
   b) $(\mathbf{Z},\leqslant)$
   c) $(P(\mathbf{Z}),\subseteq)$
   d) $(\mathbf{Z}^+$,“是倍数”)

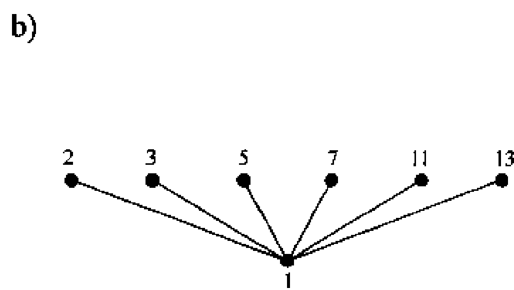
9. a) 例如，{0}和{1}
   b) 例如，4 和 6

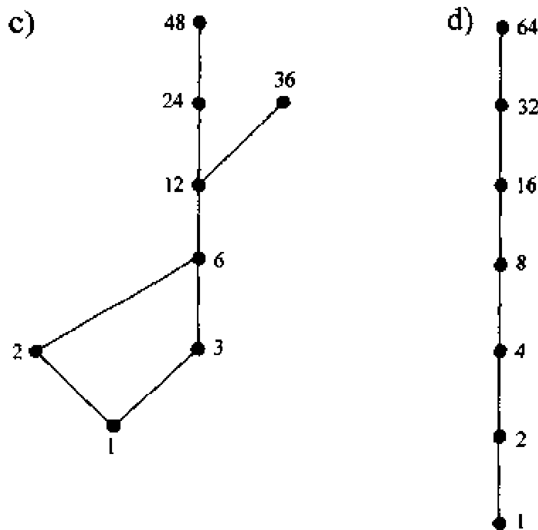
11. a) $(1,1,2)<(1,2,1)$
    b) $(0,1,2,3)<(0,1,3,2)$
    c) $(0,1,1,1,0)<(1,0,1,0,1)$

13. $0<0001<001<01<010<0101<011<11$

15.

a) 

b) 



17. (a,b),(a,c),(a,d),(b,c),(b,d),(a,a),(b,b),(c,c),(d,d)

19. (a,a),(a,g),(a,d),(a,e),(a,f),(b,b),(b,g),(b,d),(b,e),(b,f),(c,c),(c,g),(c,d),(c,e),(c,f),(g,d),(g,e),(g,f),(g,g),(d,d),(e,e),(f,f)

21. $(\varnothing,\{a\}),\{\varnothing,\{b\}\},\{\varnothing,\{c\}\},(\{a\},\{a,b\}),(\{a\},\{a,c\}),(\{b\},\{a,b\}),(\{b\},\{b,c\}),(\{c\},\{a,c\}),(\{c\},\{b,c\}),(\{a,b\}\{a,b,c\}),(\{a,c\},\{a,b,c\}),(\{b,c\},\{a,b,c\})$

23. 设$(S,\preccurlyeq)$是有穷偏序集。我们将证明这个偏序集是它的逆关系的自反传递闭包。假设(a,b)属于逆关系的自反传递闭包。那么 $a=b$ 或$a\prec b$，因此 $a\preccurlyeq b$；或者存在一个序列 a_1，a_2，…，a_n 使得$a\prec a_1\prec a_2\prec\cdots\prec a_n\prec b$，在这种情况下由$\preccurlyeq$的传递性也有 $a\preccurlyeq b$。反之，假设 $a\prec b$，如果 $a=b$，那么(a,b)属于逆关系的自反传递闭包。如果 $a\prec b$ 并且不存在z 使得$a\prec z\prec b$，那么(a,b)属于逆关系并且因此属于它的自反传递闭包。否则，设 $a\prec a_1\prec a_2\prec\cdots\prec a_n\prec b$ 是这种形式的最长的序列（由于偏序集是有穷的，所以存在这样的序列），那么中间不会有其他元素插入，因此每个对(a,a_1)，(a_1,a_2)，…，(a_n,b)属于逆关系，从而(a,b)也属于它的自反传递闭包。

25. a) 24，45　b) 3，5　c) 不存在　d) 不存在
e) 15，45　f) 15　g) 15，5，3　h) 15

27. a) {1，2}，{1，3，4}，{2，3，4}
b) {1}，{2}，{4}　c) 不存在　d) 不存在
e) {2，4}，{2，3，4}　f) {2，4}
g) {3，4}，{4}　h) {3，4}

29. 因为$(a,b)\preccurlyeq(a,b)$，$\preccurlyeq$是自反的。如果$(a_1,a_2)\preccurlyeq(b_1,b_2)$并且$(a_1,a_2)\neq(b_1,b_2)$，那么或者$a_1\prec b_1$，或者 $a_1=b_1$ 并且 $a_2\prec b_2$。在两种情况下(b_1,b_2)都不小于等于(a_1,a_2)。因此$\preccurlyeq$是反对称的。假设$(a_1,a_2)\prec(b_1,b_2)\prec(c_1,c_2)$。那么如果 $a_1\prec$

b_1 或 $b_1 \prec c_1$，我们有 $a_1 \prec c_1$，所以$(a_1, a_2) \prec (c_1, c_2)$。但是如果 $a_1 = b_1 = c_1$，那么 $a_2 \prec b_2 \prec c_2$，这也推出$(a_1, a_2) \prec (c_1, c_2)$，于是$\prec$是传递的。

31. 因为$(s,t) \preccurlyeq (s,t)$，$\preccurlyeq$是自反的。如果$(s,t) \preccurlyeq (u,v)$并且$(u,v) \preccurlyeq (s,t)$，那么 $s \preccurlyeq u \preccurlyeq s$ 且 $t \preccurlyeq v \preccurlyeq t$；从而 $s = u$ 且 $t = v$，因此$\preceq$是反对称的。假设$(s,t) \preccurlyeq (u,v) \preccurlyeq (w,x)$，那么 $s \preccurlyeq u$，$t \preccurlyeq v$，$u \preccurlyeq w$，$v \preccurlyeq x$，从而有 $s \preccurlyeq w$ 和 $t \preccurlyeq x$，因此$(s,t) \preccurlyeq (w,x)$，于是$\preccurlyeq$是传递的。

33. a) 假设 x 是极大元并且 y 是最大元，那么 $x \preccurlyeq y$。因此 x 不小于 y，从而 $x = y$。由练习 32 a) y 是唯一的。因此 x 也是唯一的。

    b) 假设 x 是极小元并且 y 是最小元，那么 $x \succcurlyeq y$。由于 x 不大于 y，从而 $x = y$。由练习 32 b) y 是唯一的。因此 x 也是唯一的。

35. a) 是　b) 不是　c) 是

37. 使用数学归纳法。设 $P(n)$是"格的每个 n 元子集有最小上界和最大下界"。基础步骤：$P(1)$为真，因为$\{x\}$的最小上界和最大下界都是 x。归纳步骤：假设$P(n)$为真。令 S 是$n+1$ 元集。设 $x \in S$ 并且$S' = S - \{x\}$。因为 S'有 n 个元素，由归纳假设它有最小上界 y 和最大下界 a。由于这是一个格，存在元素$z = \mathrm{lub}(x,y)$和 $b = \mathrm{glb}(x,a)$。如果我们可以证明 z 是S 的最小上界和b 是S 的最大下界，那么命题得证。为证明 z 是 S 的最小上界，首先观察到如果 $W \in S$，那么 $w = x$ 或 $w \in S'$。如果 $w = x$，那么 $w \preccurlyeq z$，因为 z 是x 和 y 的最小上界。如果 $w \in S'$，那么 $w \preccurlyeq z$，这是由于 y 是S'的最小上界，故 $w \preccurlyeq y$；且由于$z = \mathrm{lub}(x,y)$，有 $y \preccurlyeq z$。为证明 z 是 S 的最小上界，假设 u 是S 的一个上界，那么元素 u 一定是 x 和 y 的一个上界，但是由于$z = \mathrm{lub}(x,y)$，从而有 $z \preccurlyeq u$。类似的论述可以证明 b 是 S 的最大下界。

39. a) 不允许

    b) 允许

    c) (私有的,{印度豹,美洲狮}),
    (受限制的,{印度豹,美洲狮}),
    (注册的,{印度豹,美洲狮}),
    (私有的,{印度豹,美洲狮,黑斑羚}),
    (受限制的,{印度豹,美洲狮,黑斑羚}),
    (注册的,{印度豹,美洲狮,黑斑羚})

    d) (非私有的,{黑斑羚,美洲狮}),
    (私有的,{黑斑羚,美洲狮}),
    (受限制的,{黑斑羚,美洲狮}),
    (非私有的,{黑斑羚}),
    (私有的,{黑斑羚}),
    (受限制的,{黑斑羚}),
    (非私有的,{美洲狮}),
    (私有的,{美洲狮}),

(受限制的,{美洲狮}),(非私有的,∅),

(私有的,∅),(受限制的,∅)

41. 设Π是集合 S 的所有划分的集合，如果划分 P_1 是 P_2 的加细，即如果 P_1 中的每个集合都是 P_2 中某个集合的子集，则 $P_1 \preccurlyeq P_2$。首先我们证明$(\Pi,\preccurlyeq)$是偏序集。由于对每个划分 P 有$P\preccurlyeq P$，$\preccurlyeq$是自反的。现在假设 $P_1\preccurlyeq P_2$ 并且 $P_2\preccurlyeq P_1$。令 $T\in P_1$，因为 $P_1\preccurlyeq P_2$，存在集合 $T'\in P_2$ 使得 $T\subseteq T'$；又因为 $P_2\preccurlyeq P_1$，存在集合 $T''\in P_1$ 使得 $T'\subseteq T''$，从而 $T\subseteq T''$。但是因为 P_1 是划分，由 $T=T''$和$T\subseteq T'\subseteq T''$推出$T=T'$，于是 $T\in P_2$。反之，通过交换 P_1 与 P_2 同样得出 P_2 的每个子集也在 P_1 中。因此 $P_1=P_2$ 并且$\preccurlyeq$是反对称的。下面假设 $P_1\preccurlyeq P_2$ 并且 $P_2\preccurlyeq P_3$。设 $T\in P_1$，那么存在集合 $T'\in P_2$ 使得 $T\subseteq T'$。由于 $P_2\preccurlyeq P_3$，存在集合 $T''\in P_3$ 使得 $T'\subseteq T''$，从而有 $T\subseteq T''$，因此 $P_1\preccurlyeq P_3$，即$\preccurlyeq$是传递的。划分 P_1 和 P_2 的最大下界是划分 P，P 的子集都是形如 $T_1\cap T_2$ 的非空集合，其中 $T_1\in P_1$ 且 $T_2\in P_2$。关于这个结论的理由不再赘述。划分 P_1 与 P_2 的最小上界是对应于下述等价关系的划分：$x\in S$ 等价于 $y\in S$，如果对某个非负整数 n 存在序列 $x=x_0$，x_1，x_2，…，$x_n=y$，使得从 1 到 n 的每个 i，x_{i-1}和 x_i 在 P_1 或者 P_2 的同一个元素中。我们不再详细证明这是一个等价关系，也不再证明它就是两个划分的最小上界。

43. 由练习 37 对整个有穷格存在最小上界和最大下界。根据定义这些元素分别是最大元和最小元。

45. $\mathbf{Z}^+\times\mathbf{Z}^+$的子集的最小元是具有最小的横坐标的对，并且如果存在多个这样的对，它就是这些对中具有最小的纵坐标的对。

47. $a\prec b\prec c\prec d\prec e\prec f\prec g\prec h\prec i\prec j\prec g\prec k\prec l\prec m$

49. $C\prec A\prec B\prec D\prec E\prec F\prec G$

补充练习

1. a) 反自反的(这里不包含空串)，对称的
   b) 反自反的,对称的
   c) 反自反的,反对称的,传递的

3. 因为 $a+b=a+b$，$((a,b),(a,b))\in R$，因此 R 是自反的。如果$((a,b),(c,d))\in R$，那么 $a+d=b+c$，因此 $c+b=d+a$。从而$((c,d),(a,b))\in R$，于是 R 是对称的。假设$((a,b),(c,d))$和$((c,d),(e,f))$属于 R，那么 $a+d=b+c$ 且 $c+f=d+e$。把这两个等式相加，然后从两边减去 $c+d$ 得到$a+f=b+e$。从而$((a,b),(e,f))$属于 R，于是 R 是传递的。

5. 假设$(a,b)\in R$。因为$(b,b)\in R$，从而有$(a,b)\in R^2$。

7. 是，是

9. 是，是

11. 在投影中具有相同键码的两个记录在原来的关系中有相同的键码。

13. $(\Delta\cup R)^{-1}=\Delta^{-1}\cup R^{-1}=\Delta\cup R^{-1}$

15. a) $R=\{(a,b),(a,c)\}$。R 的对称闭包的传递闭包是$\{(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)\}$，它与 R 的传递闭包的对称闭包是不同的，后者是$\{(a,b),(a,c),(b,a),(c,a)\}$。

    b) 假设(a,b)属于 R 的传递闭包的对称闭包。必须证明(a,b)属于 R 的对称闭包的传递闭包。我们知道(a,b)和(b,a)中至少有一个属于 R 的传递闭包，因此在 R 中存在一条从 a 到 b 或者从 b 到 a 的路径（或者两条都有）。在前面的情况下，在 R 的对称闭包中存在一条从 a 到 b 的路径。在后面的情况下，可以通过把从 b 到 a 路径中的所有边改变方向向回走，在 R 的对称闭包中得到一条从 a 到 b 的路径。因此(a,b)属于 R 的对称闭包的传递闭包。

17. 由于 $R\subseteq S$，S 的关于性质 **P** 的闭包是包含 R 的具有性质 **P** 的关系。因此 S 的关于性质 **P** 的闭包包含了 R 关于性质 **P** 的闭包。

19. 使用沃舍尔（Warshall）算法的基本思想，不同的只是令 $w_{ij}^{[k]}$等于使用下标不超过 k 的内部顶点的从 v_i 到 v_j 的最长路径的长度。并且如果没有这种路径，令 $w_{ij}^{[k]}$等于 -1。为了从 $\mathbf{W}_{k-1}$的元素中找到 $w_{ij}^{[k]}$，对于每个(i,j)对确定是否存在不使用序标大于 k 的顶点的从 v_i 到 v_k 的路径和从 v_k 到 v_j 的路径。如果 $w_{ik}^{[k-1]}$或 $w_{kj}^{[k-1]}$是 -1，那么这对路径不存在，因此置 $w_{ij}^{[k]}=w_{ij}^{[k-1]}$。如果这对路径存在，那么存在两种可能。如果 $w_{kk}^{[k-1]}>0$，那么存在任意长的从 v_i 到 v_j 的路径，因此置 $w_{ij}^{[k]}=\infty$；如果 $w_{kk}^{[k-1]}=0$，置 $w_{ij}^{[k]}=\max(w_{ij}^{[k-1]},w_{ik}^{[k-1]}+w_{kj}^{[k-1]})$。（初始取 $\mathbf{W}_0=\mathbf{M}_R$。）

21. 25

23. 因为 $A_i\cap B_j$ 是 A_i 和 B_j 的子集，则这些子集的集合是每个给定划分的加细。我们必须证明这是一个划分。根据构造，每个这样的集合是非空的。下面说明这些集合的并就是 S。假设 $s\in S$。因为 P_1 和 P_2 是 S 的划分，存在集合 A_i 和 B_j 使得 $s\cap A_i$ 且 $s\in B_j$，从而 $s\in A_i\cap B_j$。因此这些集合的并就是 S。再观察到除非 $i=i'$，$j=j'$，$(A_i\cap B_j)\cap(A_{i'}\cap B_{j'})=(A_i\cap A_{i'})\cap(B_j\cap B_{j'})=\varnothing$。

25. 子集关系是任何集合族上的偏序，因为它是自反的，反对称的、传递的，这里的集合族是**R**(S)。

27. 确定用户需求＜写出功能规约＜设置检测点＜开发系统规约＜写文档＜开发模块 A＜开发模块＜B＜开发模块 C＜模块集成＜α 测试＜β 测试＜完成

29. a) 具有多个元素的反链只有$\{c,d\}$。

    b) 具有多个元素的反链只有$\{b,c\}$,$\{c,e\}$和$\{d,e\}$。

    c) 具有多个元素的反链只有$\{a,b\}$,$\{a,c\}$,$\{b,c\}$,$\{a,b,c\}$,$\{d,e\}$,$\{d,f\}$,$\{e,f\}$,$\{d,e,f\}$。

31. 设$(S,\preccurlyeq)$是有穷偏序集，且 A 是一条极大链。因为$(A,\preccurlyeq)$也是一个偏序集，它一定有极小元 m。假设 m 不是 S 中的极小元，那么存在 S 中的元素 a 使得 $a\prec m$。但是，这就使得集合 $A\cup\{a\}$变成比 A 更大的链，为此我们需要证明 a 与 A 中每一个元素都可比。因为 m 与 A 中每个元素都可比，并且 m 是极小元，当 x 属于 A 且 $x\neq m$ 时有 $m\prec x$。由于 $a\prec m$ 且 $m\prec x$，由传递性可得对于 A 中的每个元素 x 有 $a\prec x$。

33. 令 aRb 表示 a 是 b 的后代。由练习 32，如果不存在 $n+1$ 个人的集合使得其中每个人都不是其他人（一条反链）的后代，那么 $k \leqslant n$。因此这个集合可以被划分成 $k \leqslant n$ 条链。根据鸽巢原理，这些链中至少有一条链包含至少 $m+1$ 个人。

35. 基础步骤：$a_{0,0}=[0(0+1)/2]+0=0$。归纳步骤：假设对所有小于(m,n)的对为真。如果 $n=0$，那么由于$(m-1,n)\prec(m,n)$，$a_{m,n}=a_{m-1,n}+1=[n(n+1)/2]+m-1+1=[n(n+1)/2]+m$，正如所求。如果 $n\neq 0$，那么由于$(m,n-1)\prec(m,n)$，$a_{m,n}=a_{m,n-1}+n=[(n-1)(n-1+1)/2]+m+n=[n(n+1)/2]+m$。

37. 假设 R 是近似序。因为 R 是自反的，如果 $a\in A$，那么有$(a,a)\in R$，从而有$(a,a)\in R^{-1}$。因此$(a,a)\in R\cap R^{-1}$。于是 $R\cap R^{-1}$是自反的。对任何关系 R，$R\cap R^{-1}$是对称的，因为对任何关系 R，如果$(a,b)\in R$，那么有$(b,a)\in R^{-1}$；反之也对。为证明 $R\cap R^{-1}$是传递的，假设$(a,b)\in R\cap R^{-1}$并且$(b,a)\in R\cap R^{-1}$。因为 R 是传递的，由$(a,b)\in R$ 和$(b,c)\in R$ 推出$(a,c)\in R$。类似地，由$(a,b)\in R^{-1}$和$(b,c)\in R^{-1}$，就有$(b,a)\in R$ 和$(c,b)\in R$，因此$(c,a)\in R$，$(a,c)\in R^{-1}$。于是$(a,c)\in R\cap R^{-1}$。从而证明了 $R\cap R^{-1}$是等价关系。

39. a) 因为$\mathrm{glb}(x,y)=\mathrm{glb}(y,x)$和$\mathrm{lub}(x, y)=\mathrm{lub}(y, x)$，从而有 $x\wedge y=y\wedge x$ 和 $x\vee y=y\vee x$。
   b) 使用定义，$(x\wedge y)\wedge z$ 是x，y 与 z 的下界，且大于每一个其他的下界。因为 x，y 和 z 是可互换的，$x\wedge(y\wedge z)$ 是同一元素。类似地，$(x\vee y)\vee z$ 是x，y 与 z 的上界，并且小于每一个其他的上界。因为 x，y 和 z 是可互换的，$x\vee(y\vee z)$也是同一元素。
   c) 为证明$x\wedge(x\vee y)=x$，只需证明 x 是x 与 $x\vee y$ 的最大下界。观察到 x 是 x 的下界，根据定义 $x\vee y$ 大于x，x 也是它的下界。所以 x 是x 与$x\vee y$ 的下界。但是 x 的任何下界必须小于等于x。因此 x 是最大的下界。第二个公式是第一个公式的对偶式，证明省略。
   d) x 是自身的下界，也是自身的上界，并且也是最大的下界，最小的上界。

41. a) 因为 1 是大于等于 1 的唯一元素，因此它也是 1 的唯一上界，因而也是 x 和 1 的最小上界的唯一可能的值。
   b) 因为 $x\leqslant 1$，x 是x 和 1 的下界并且没有其他的下界可能大于 x，因此 $x\wedge 1=x$。
   c) 因为 $0\leqslant x$，x 是x 和 0 的上界并且没有其他的上界可能小于 x，因此 $x\vee 0=x$。
   d) 因为 0 是小于等于 0 的唯一元素，因此它也是 0 的唯一下界，因而也是 x 是 0 的最大下界的唯一可能的值。

43. $L=(S,\subseteq)$，其中$S=\{\varnothing,\{1\},\{2\},\{3\},\{1,2\},\{2,3\},\{1,2,3\}\}$

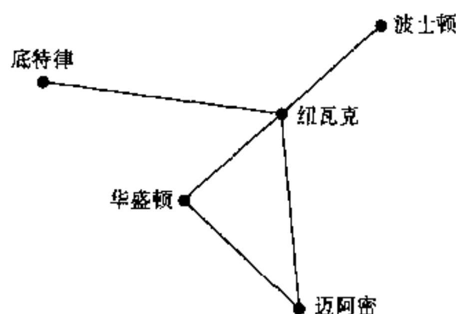
45. 是

47. 子集 $X\subseteq S$ 的补是它的补集 $S-X$。为证明这一点，注意到因为$X\cup(S-X)=S$ 和 $X\cap(S-X)=\varnothing$，因此$X\cup(S-X)=1$ 且 $X\cap(S-X)=0$。

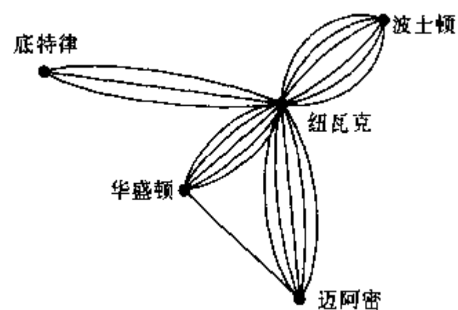
第 7 章

7.1 节

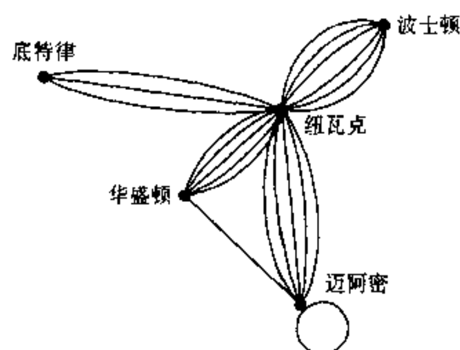
1.　a)



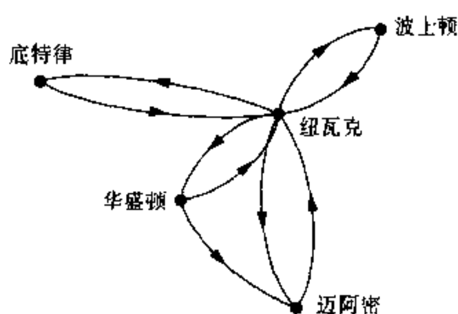
b)



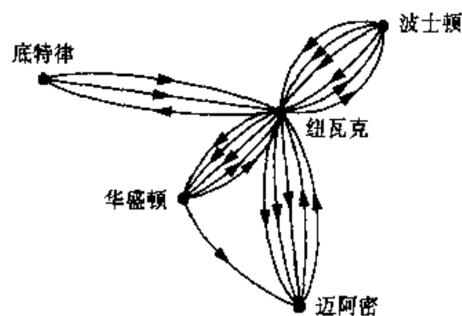
c)



d)

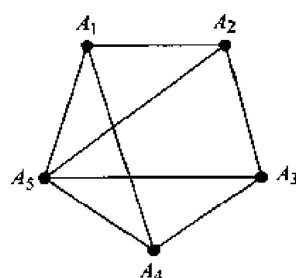


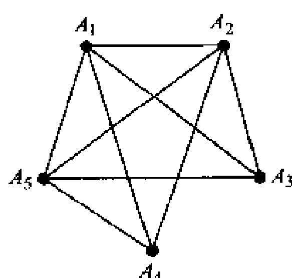
e)

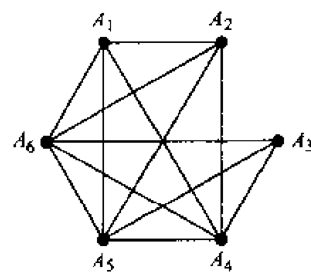


3. 简单图
5. 伪图
7. 有向图
9. 有向多重图

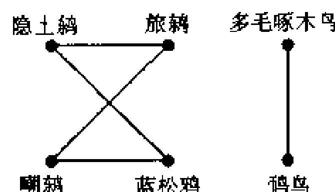
11. a) b) c)



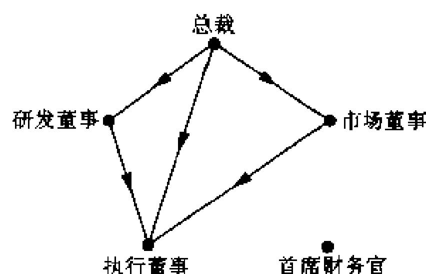




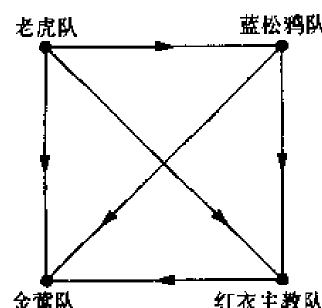
13.



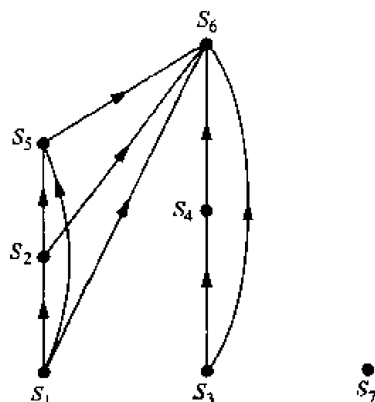
15.



17.



19.



21. 用顶点表示组里的人们。对每对顶点来说，都在图里放置一条有向边。对从表示 A 的顶点到表示 B 的顶点的边来说，若 A 喜欢 B，则用 +（加）标记；若 A 不喜欢 B，则用 −（减）标记；若 A 对 B 持中立态度，则用 0 标记。

7.2 节

1. $v=6$; $e=6$; $\deg(a)=2$，$\deg(b)=4$，$\deg(c)=1$，$\deg(d)=0$，$\deg(e)=2$，$\deg(f)=3$；c 是悬挂点；d 是孤立点。

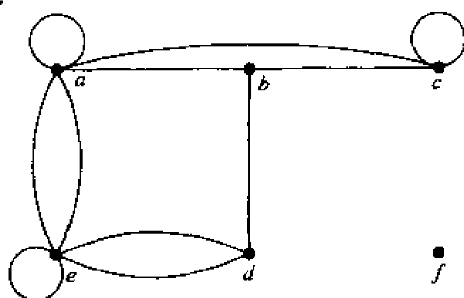
3. $v=9$; $e=12$; $\deg(a)=3$，$\deg(b)=2$，$\deg(c)=4$，$\deg(d)=0$，$\deg(e)=6$，$\deg(f)=0$；$\deg(g)=4$；$\deg(h)=2$；$\deg(i)=3$；d 和 f 都是孤立点。

5. 否，因为顶点的度之和不可能是奇数。

7. $v=4$; $e=7$; $\deg^-(a)=3$, $\deg^-(b)=1$, $\deg^-(c)=2$, $\deg^-(d)=1$, $\deg^+(a)=1$, $\deg^+(b)=2$, $\deg^+(c)=1$, $\deg^+(d)=3$。

9. 5 个顶点，13 条边；$\deg^-(a)=6$, $\deg^+(a)=1$, $\deg^-(b)=1$, $\deg^+(b)=5$, $\deg^-(c)=2$, $\deg^+(c)=5$, $\deg^-(d)=4$, $\deg^+(d)=2$, $\deg^-(e)=0$, $\deg^+(e)=0$。

11.



13. 偶图

15. 非偶图

17. 非偶图

19. a) n 个顶点，$n(n-1)/2$ 条边

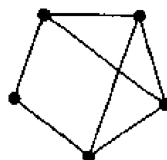
b) n 个顶点，n 条边

c) $n+1$ 个顶点，$2n$ 条边

d) $m+n$ 个顶点，mn 条边

e) 2^n 个顶点，$n2^{n-1}$条边

21. a) 是

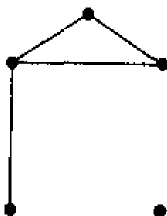


b) 否，度之和是奇数。

c) 否

d) 否，度之和是奇数。

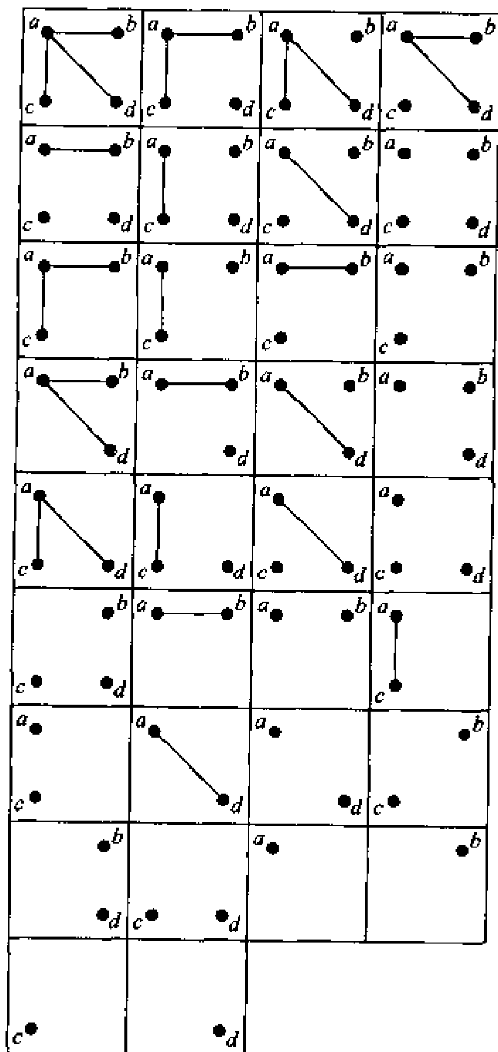
e) 是



f) 否，度之和是奇数。

23. 17

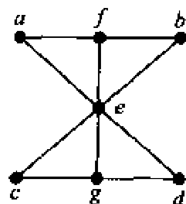
25.



27. a) 对所有 $n\geqslant 1$ b) 对所有 $n\geqslant 3$ c) 对 $n=3$ d) 对所有 $n\geqslant 0$

29. 5

31.



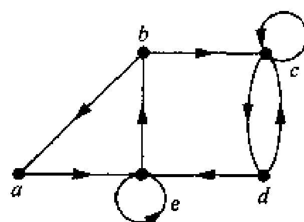
33. a) 带 n 个顶点而且没有边的图
    b) K_m 和 K_n 的不相交并图
    c) 带有顶点 $\{v_1, \cdots, v_n\}$，且在 v_i 与 v_j 之间有边（除非 $i \equiv j \pm 1 \pmod n$）的图
    d) 用长度为 n 的位串表示其顶点，若两个顶点所对应的位串相差超过一位，则在这两个顶点之间有一条边的图

35. $v(v-1)/2 - e$

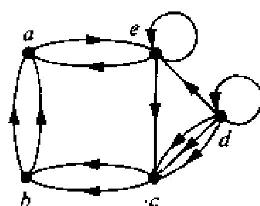
37. G 和 $\bar{G}$ 的并图包含 n 个顶点中每对顶点之间的一条边。因此这个并图是 K_n。

39. 7.1 节练习

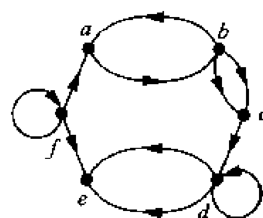
练习7:



练习8:

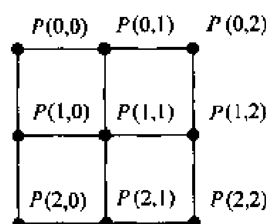


练习9:



41. 有向图 $G = (V, E)$ 是它自身的逆图当且仅当它满足条件：$(u, v) \in E$ 当且仅当 $(v, u) \in E$。但这恰好是这样的条件：所对应的关系必定是对称的。

43.



45. 可以这样连接 $P(i, j)$ 与 $P(k, l)$：用 $|i-k|$ 个 hop 连接 $P(i, j)$ 与 $P(k, j)$ 用 $|j-l|$ 个 hop 连接 $P(k, j)$ 与 $P(k, l)$。因此连接 $P(i, j)$ 与 $P(k, l)$ 所需要的 hop 总数不超过 $|i-k| + |j-l|$。这个值小于或等于 $m + m = 2m$，即 $O(m)$。

7.3 节

1.

顶点	相邻顶点
a	b，c，d
b	a，d
c	a，d
d	a，b，c

3.

顶点	终结顶点
a	a，b，c，d
b	d
c	a，b
d	b，c，d

5.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

其中顶点都是以字母表顺序排列的

7.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

9. a)

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

b)

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

c)

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

d)

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

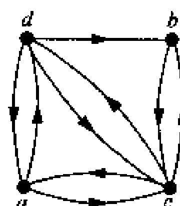
e)

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

f)

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

11.



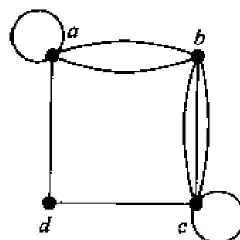
13.

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

15.

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

17.



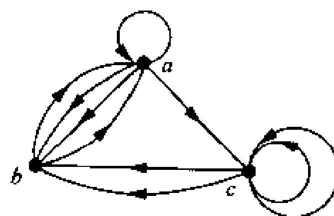
19.

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

21.

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

23.



25. 是

27. 练习 13：

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

练习 14：

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

练习 15：

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

29. $\deg(v)$ − 在 v 处的环数；$\deg^{-}(v)$

31. 若 e 不是环则是 2，若 e 是环则是 1。

33. a) $\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 1 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ & & & & 1 & 0 & \cdots & 0 \\ & \mathbf{B} & & & 0 & 1 & \cdots & 0 \\ & & & & \vdots & \vdots & & \vdots \\ & & & & 0 & 0 & \cdots & 1 \end{bmatrix}$，其中 **B** 是 b) 的答案

d) $\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ 1 & 0 & & 0 & 1 & \cdots & 0 \\ 0 & 1 & & 0 & 0 & & \\ \vdots & \vdots & & \vdots & \vdots & & \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 1 \end{bmatrix}$

35. 同构

37. 同构

39. 同构

41. 不同构

43. 同构

45. G 是通过恒等函数来同构于它自身的，所以同构是自反的。假设 G 是同构于 H 的。于是存在保持相邻性与非相邻性的从 G 到 H 的一一对应 f。所以 f^{-1} 是保持相邻性与非相邻性的从 H 到 G 的一一对应。因此同构是对称的。若 G 是同构于 H 的，且 H 是同构于 K 的，则存在保持相邻性与非相邻性的从 G 到 H 和从 H 到 K 的一一对应 f 和 g。所以 $g \circ f$ 是保持相邻性与非相邻性的从 G 到 K 的一一对应。因此同构是传递的。

47. 全部是零
49. 按顺序标记顶点，使得顶点集划分的第 1 个集合里的顶点都排在前面。因为没有边连接划分的同一个集合里的顶点，所以这个矩阵具有所需要的形式。
51. C_5
53. 只有 $n=5$
55. 4
57. a) 是　　b) 否　　c) 否
59. $G=(V_1,E_1)$是同构于 $H=(V_2,E_2)$的，当且仅当存在从 V_1 到 V_2 的函数 f 和从 E_1 到 E_2 的函数 G，使得每个函数都是一一对应的；并且对 E_1 里的每一条边 e 来说，$g(e)$ 的端点是 $f(v)$和 $f(w)$，其中 v 和 w 是 e 的端点。
61. 是
63. 是
65. 若 f 是从有向图 G 到有向图 H 的同构，则 f 也是从 G^c 到 H^c 的同构。为证明这一点，注意(u,v)是 G^c 的边当且仅当(v,u)是 G 的边，当且仅当$(f(v),f(u))$是 H 的边，当且仅当$(f(u),f(v))$是 H^c 的边。
67. 乘积是 $[a_{ij}]$，其中当 $i\neq j$ 时，a_{ij}是从 v_i 到 v_j 的边数，而 a_{ii}是与 v_i 关联的边数。
69. 练习 41 里的图提供了一个魔鬼对。

7.4 节

1. a) 长度为 4 的通路；不是通路；不是简单通路
   b) 不是通路
   c) 不是通路
   d) 长度为 5 的简单通路
3. 否
5. 否
7. a) 2　　b) 7　　c) 20　　d) 61
9. a) 3　　b) 0　　c) 27　　d) 0
11. a) 1　　b) 0　　c) 2　　d) 1　　e) 5　　f) 3
13. 根据定义 R 是自反的。假设$(u,v)\in R$，则存在从 u 到 v 的通路。于是$(v,u)\in R$，因为存在从 v 到 u 的通路，即把从 u 到 v 的通路反过来。假设$(u,v)\in R$ 和$(v,w)\in R$，则存在从 u 到 v 和从 v 到 w 的通路。把这两条通路连接起来就给出从 u 到 w 的通路。因此$(u,w)\in R$。所以 R 是传递的。
15. c
17. b，c，e，i
19. 若一个顶点是悬挂点，则它显然不是割点。所以割边的端点要是割点，它就不是悬挂点。删除一条割边会产生出比原来的图有更多连通分支的图。若割边的端点不是悬挂点，则删除这条割边后，这端点所在的连通分支里包含除这个顶点外的更多顶点。所以，删除那个顶点以及与它关联的所有边，包括原来的割边，会产生出比原来的图有更多连通分支的图。因此割边的端点要不是悬挂点，它就是割点。

21. 假设存在带至多一个不是割点的顶点的连通图 G。顶点 u 和 v 之间的距离定义成在 G 里 u 和 v 之间最短通路的长度，表示成 $d(u,v)$。设 s 和 t 是使得 $d(s,t)$ 达到最大的 G 里的顶点。s 或者 t（或二者全部）是割点。所以不失一般性，假设 s 是割点。在通过从 G 删除 s 和与 s 关联的所有边而得出的图里，设 w 属于不包含 t 的那个连通分支。因为从 w 到 t 的每条通路都包含 s，所以 $d(w,t)>d(s,t)$，矛盾。

23. a) 丹佛－芝加哥，波士顿－纽约

    b) 西雅图－波特兰，波特兰－旧金山，盐湖城－丹佛，纽约－波士顿，波士顿－伯灵顿，波士顿－班哥尔。

25. 联合起来（间接地或直接地）影响到每一个人的一组人；{Deborah，Yvonne}

27. 一条边不可能连接两个在不同连通分支里的顶点。因为在带 n_i 个顶点的连通分支里至多有 $C(n_i,2)$ 条边，所以在图中至多有 $\sum_{i=1}^{k} C(n_i,2)$ 条边。

29. 假设 G 是不连通的。于是对满足 $1 \leqslant k \leqslant n-1$ 的某个 k 来说，它有带 k 个顶点的一个分支。G 所可能有的边至多是 $C(k,2)+C(n-k,2)=(k(k-1)+(n-k)(n-k-1))/2=k^2-nk+(n^2-n)/2$。这个 f 的二次函数在 $k=n/2$ 上达到最小，而在 $k=1$ 或 $k=n-1$ 上达到最大。因此若 G 不是连通的，则边数不会超过该函数在 1 和在 $n-1$ 上的值，即 $(n-1)(n-2)/2$。

31. a) 1　　b) 2　　c) 6　　d) 21

33. 2

35. 设通路 P_1 和 P_2 分别是 $u=x_0$，x_1，…，$x_n=v$。因为 P_1 和 P_2 不包含相同的边的集合，所以它们必然逐渐分叉。若这个分叉发生在它们中的一条已经结束之后，则另一条通路的剩余部分是从 v 到 v 的简单回路。否则，可以假设 $x_0=y_0$，$x_1=y_1$，…，$x_i=y_i$，但是 $x_{i+1}\neq y_{i+1}$。沿着通路 y_i，y_{i+1}，y_{i+2} 等等前进，直到它再次遇到 P_1 上的顶点为止。一旦回到 P_1 上，就沿着它，在必要时向前或向后，回到 X_i。因为 $x_i=y_i$，这样就形成一条回路，它必然是简单的，因为在这些 x_k 之间没有一条边能等于用过的 y_i 之间的一条边。

37. 图 G 是连通的当且仅当 $\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots+\mathbf{A}^{n-1}$ 的每个非对角线项都是正的，其中 $\mathbf{A}$ 是 G 的邻接矩阵。

7.5 节

1. 否

3. 否

5. a，b，c，d，c，e，d，b，e，a，e，a

7. a，i，h，g，d，e，f，g，c，e，h，d，c，a，b，i，c，b，h，a

9. 存在欧拉通路：f，a，b，c，d，e，f，b，d 是一条这样的欧拉通路。

11. 存在欧拉通路：b，c，d，e，f，d，g，i，d，a，h，i，a，b，i，c 是一条这样的欧拉通路。

13. 存在欧拉通路：b，c，d，e，f，d，g，i，d，a，h，i，a，b，i，c

15. 否，A 仍然有奇数的度。

17. 当顶点表示交叉路口而边表示街道的图有欧拉通路时

e :=在 H 里带端点 V 的第一条边，若这样的边存在（相对于 V 的列表而言），使得 e 不是 H 的割边；否则就是在 H 里带端点 V 的第一条边

w := e 的另一个端点

circuit := *circuit* 添加上边 e 和 w

v := w

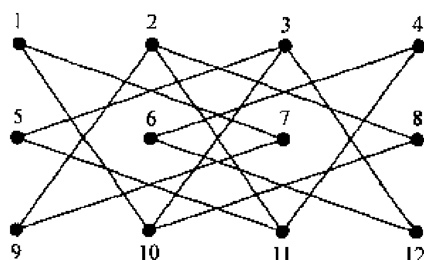
H := $H - e$

end {*circuit* 是一条欧拉回路}

61. 若 G 有欧拉回路，则它也有欧拉通路。若它没有，则在有奇数度的两个顶点之间添加一条边，并且应用这个算法来得到欧拉回路，然后删除这条新的边。

63. 假设 $G=(V,E)$ 是满足 $V=V_1\cup V_2$ 的偶图，其中没有边是连接 V_1 里的顶点与 V_2 里的顶点的。假设 G 有哈密顿回路。这样的回路必然是形如 a_1，b_1，a_2，b_2，…，a_k，b_k，a_1，b_1，其中对 $i=1$，2，…，k 来说，有 $a_i\in V_1$ 和 $b_i\in V_2$。因为哈密顿回路访问每个顶点恰好一次，所以除了回路开始和结束的 v_1 之外，所有图中的顶点数等于 $2k$，是偶数。因此，带奇数个顶点的偶图不可能有哈密顿回路。

65.



67. 把 3×4 棋盘的格子表示如下：

1	2	3	4
5	6	7	8
9	10	11	12

马的巡回路线可以通过下列移动来构成：8，10，1，7，9，2，11，5，3，12，6，4。

69. 把 4×4 棋盘的格子表示如下：

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

角上的四个格子里只有两种移动。若包含1－10，1－7，16－10和16－7的所有边，则过早地完成了回路，所以必须去掉这些边中的至少一条。不失一般性，假设通路开始于1－10，10－16和16－7。现在格子3仅有的移动是到格子5，10和12，而格子10已经有了两条关联的边。因此3－5和3－12必须是在哈密顿回路里。同理，边8－2和8－15必须是在回路里。现在格子9仅有的移动是到格子2，7，和15。假如有从格子9到格子2和15的边，则过早地完成了回路。因此边9－7必须是在回路里，让格子7关联的边达到饱和。但是现在格子14被迫连接到格子5和12，从而过早地完成了回路（5－14－12－3－5）。这个矛盾证明在4×4棋盘上没有马的巡回路线。

71. 因为在 $m\times n$ 棋盘上有 mn 个格子，所以若 m 和 n 都是奇数，则有奇数个格子。因为根据练习70，对应的图是偶图，所以根据练习63，它没有哈密顿回路。因此没有可重入的马的巡回路线。

7.6 节

1. a) 顶点是车站，边连接相邻的车站，权是在相邻车站之间旅行所需要的时间。

b）与 a）相同，不同的是权是在相邻车站之间的距离。

c）与 a）相同，不同的是权是在相邻车站之间的票价。

3. 16

5. 练习 2：a，b，e，d，z

练习 3：a，c，d，e，g，z

练习 4：a，b，e，h，l，m，p，s，z

7. a）a，c，d　　b）a，c，d，f　　c）c，d，f　　d）b，d，e，g，z

9. a）直达　　b）经过纽约　　c）经过亚特兰大和芝加哥　　d）经过纽约

11. a）经过芝加哥　　b）经过芝加哥　　c）经过洛杉矶　　d）经过芝加哥

13. a）经过芝加哥　　b）经过芝加哥　　c）经过洛杉矶　　d）经过芝加哥

15. 当把 z 添加到集合 S 时不停止算法。

17. a）经过木桥，经过木桥和康登

b）经过木桥，经过木桥和康登

19. 例如，观光旅游、街道清扫

21.

	a	b	c	d	e	z
a	4	3	2	8	10	13
b	3	2	1	5	7	10
c	2	1	2	6	8	11
d	8	5	6	4	2	5
e	10	7	8	2	4	3
z	13	10	11	5	3	6

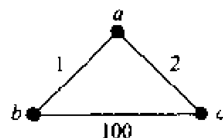
23. $O(n^3)$

25. $a-c-b-d-a$（或从某个顶点开始，但是以相同或相反的方向来经过各项点的相同的回路）

27. 旧金山—丹佛—底特律—纽约—洛杉矶—旧金山（或从某个顶点开始，但是以相同或相反的方向来经过各顶点的相同的回路）

29. 考虑右面的图：

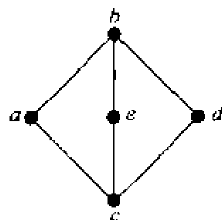
回路 $a-b-a-c-a$ 访问每个顶点至少一次（访问顶点 a 两次），并且总的权为 6。每个哈密顿回路的总权为 103。



7.7 节

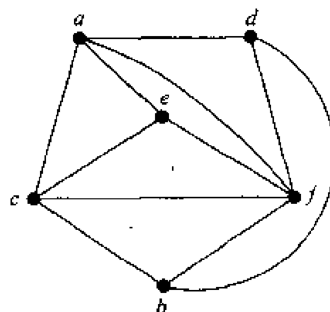
1. 是

3.



5. 否

7. 是



9. 通过包含连接 v_1，v_2 和 v_3 的边的 K_5 的子图的平面表示来形成一个三角形。顶点 v_4 必须放置在这个三角形之内或之外。这里只考虑当 v_4 是在这个三角形之内时的情形；另外一种情形是类似的。画出从 v_4 到 v_1，v_2 和 v_3 的三条边就形成四个面。无论 v_5 是在这四个面中的哪个，它都只可能连接到其他顶点中的三个，而不是所有四个。

11. 8

13. 因为没有环和多重边，也没有长度为 3 的简单回路，而且无界的面的次数至少为 4，所以每个面的次数都至少为 4。因此 $2e \geq 4r$，或 $r \leq e/2$。但是 $r = e - v + 2$，所以有 $e - v + 2 \leq e/2$，它蕴涵着 $e \leq 2v - 4$。

15. 正如推论 2 里那样，有 $2e \geq 5r$ 和 $r = e - v + 2$。因此 $e - v + 2 \leq 2e/5$，它蕴涵着 $e \leq (5/3)v - (10/3)$。

17. 只有 a）和 c）

19. 不同构于 $K_{3,3}$

21. 平面性的

23. 非平面性的

25. a）1　b）3　c）9　d）2　e）4　f）16

27. 照提示里所描述的那样画出 $K_{m,n}$。交叉数是在第一象限里的数目的 4 倍。在 x 轴上原点右方的顶点是（1，0），（2，0），…，（$m/2$，0），而在 y 轴上原点上方的顶点是（0，1），（0，2），…，（0，$n/2$）。通过选择满足 $1 \leq a < b \leq m/2$ 的两个不同的数 a 和 b，以及现在满足 $1 \leq r < s \leq n/2$ 的两个不同的数 r 和 s，就得到所有的交叉；得到连接（a，0）和（0，s）的边与连接（b，0）和（0，r）的边之间的恰好一个交叉。因此在第一象限里的交叉数是

$$C\left(\frac{m}{2},2\right)\cdot C\left(\frac{n}{2},2\right)=\frac{(m/2)(m/2-1)}{2}\cdot\frac{(n/2)(n/2-1)}{2}$$

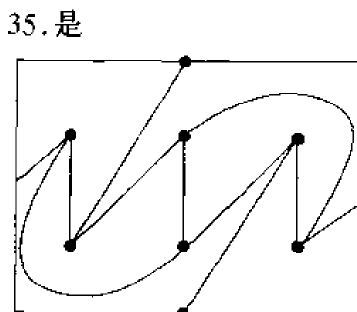
因此总的交叉数是 $4 \cdot mn(m-2)(n-2)/64 = mn(m-2)(n-2)/16$

29. a）2　b）2　c）2　d）2　e）2　f）2

31. 对 $n \leq 4$ 来说这个公式是有效的。若 $n > 4$，则根据练习 30，K_n 的厚度至少是 $C(n,2)/(3n-6) = (n+1+2/(n-2))/6$ 向上取整。因为这个值永远不是整数，所以它等于向下取整的下一个最大整数，即 $\lfloor (n+7)/6 \rfloor$。

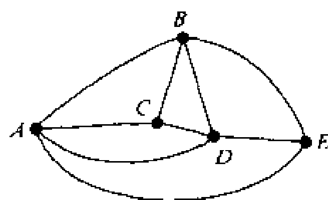
33. 这个结果是从练习 32 得出的，因为 $K_{m,n}$ 有 mn 条边和 $m+n$ 个顶点，而没有三角形，因为它是偶图。

35. 是



7.8 节

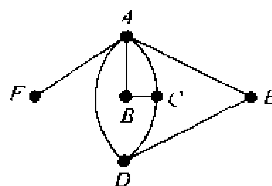
1. a）



b）



c）



3. 3
5. 3
7. 2
9. 3
11. 没有边的图
13. 若 n 是偶数则为 3，若 n 是奇数则为 4。
15. 时间段 1：Math 115，Math 185；时间段 2：Math 116，CS 473；时间段 3：Math 195，CS 101；时间段 4：CS 102；时间段 5：CS 273
17. 5
19. 练习 3：3
    练习 4：6
    练习 5：3
    练习 6：4
    练习 7：3
    练习 8：6
    练习 9：4
21. 5
23. 带其中一种颜色的顶点的集合是一个部分，带另外一种颜色的顶点的集合是另外一个部分。因为没有边可以连接相同颜色的顶点，所以在同一个部分里的顶点之间没有边。
25. 颜色 1：e，f，d；颜色 2：c，a，i，g；颜色 3：h，b，j。
27. 着色 C_6
29. a) 6　　b) 7　　c) 9　　d) 11
31. 用颜色表示频率并且用顶点表示区域。若两个顶点所表示的区域有互相干涉，则用边连接这两个顶点。于是一个 k 重着色恰好就是一种避免干涉的频率分配。

补充练习

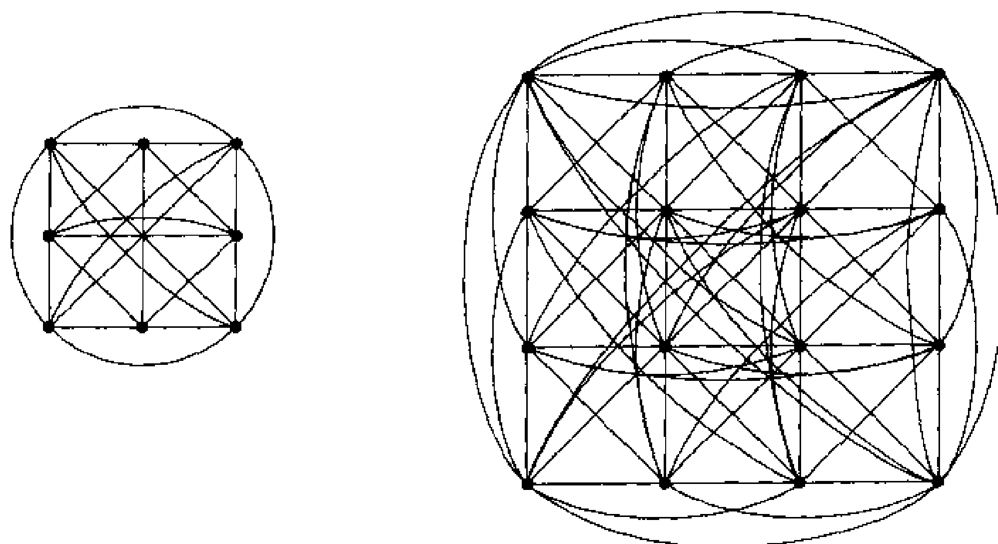
1. 2500
3. 是
5. 是
7. $\sum_{i=1}^{m} n_i$ 个顶点，$\sum_{i<j} n_i n_j$ 条边
9. a)　　b)　　c)

11. 完全子图包含下列顶点集：$\{\{b,c,e,f\},\{a,b,g\},\{a,d,g\},\{d,e,g\},\{b,e,g\}\}$
13. 完全子图包含下列顶点集：$\{\{b,c,d,j,k\},\{a,b,j,k\},\{e,f,g,i\},\{a,b,i\}\{a,i,j\},$

$\{b,d,e\}$，$\{b,e,i\}$，$\{b,i,j\}$，$\{g,h,i\}$，$\{h,i,j\}$

15. $\{c,d\}$是最小支配集

17. a) b)



19. a) 1 b) 2 c) 3

21. a) 在图 G 里从 u 到 v 的通路导出在同构的图 H 里从 $f(u)$ 到 $f(v)$ 的通路。

b) 假设 f 是从 G 到 H 的同构。若 $v_0, v_1, \cdots, v_n, v_0$ 是 G 里的哈密顿回路，则 $f(v_0), f(v_1), \cdots, f(v_n), f(v_0)$ 必然是 H 里的哈密顿回路，因为它仍然是回路并且对 $0 \leqslant i < j \leqslant n$ 来说有 $f(v_i) \neq f(v_j)$。

c) 假设 f 是从 G 到 H 的同构。若 v_0，v_1，…，v_n，v_0 是 G 里的欧拉回路，则 $f(v_0)$，$f(v_1), \cdots, f(v_n), f(v_0)$ 必然是 H 里的欧拉回路，因为它是包含每条边恰好一次的回路。

d) 两个同构的图必然有相同的交叉数，因为可以在平面里以同样的方式来画出它们。

e) 假设 f 是从 G 到 H 的同构。则 v 是 G 里的孤立点当且仅当 $f(v)$ 是 H 里的孤立点。因此这些图必然有相同的孤立点数。

f) 假设 f 是从 G 到 H 的同构。若 G 是偶图，则 G 的顶点集可以划分成 V_1 和 V_2，其中没有边连接 V_1 里的顶点与 V_2 里的顶点。于是 H 的顶点集可以划分成 $f(V_1)$ 和 $f(V_2)$，其中没有边连接 $f(V_1)$ 里的顶点与 $f(V_2)$ 里的顶点。

23. 3

25. a) 是 b) 否

27. 否

29. 是

31. 如果 e 是带端点 u 和 v 的割边，那么若让 e 的方向是从 u 到 v，则将没有从 v 到 u 的在这个有向图里的通路，否则 e 就不是割边。若让 e 的方向是从 v 到 u，则类似的推理也成立。

33. $n-1$

35. 设顶点表示这些鸡。把边 (u, v) 包含在这个图里当且仅当鸡 u 支配鸡 v。

37. a) 4　　b) 2　　c) 3　　d) 4　　e) 4　　f) 2

39. a) 假设 $G=(V, E)$。设 $a, b\in V$。必须证明在 $\bar{G}$ 里 a 和 b 之间的距离至多为 2。若 $\{a, b\}\notin E$，则这个距离为 1，所以假设 $\{a, b\}\in E$。因为 G 的直径大于 3，所以存在顶点 u 和 v，使得 u 和 v 之间的距离在 G 里大于 3。要么 u 要么 v，或者它们二者是不属于集合 $\{a, b\}$ 的。假设 u 是与 a 和 b 都不同的。要么 $\{a, u\}$ 要么 $\{b, u\}$ 是属于 E 的；否则 a, u, b 是 $\bar{G}$ 里长度为 2 的通路。所以，不失一般性，假设 $\{a, u\}\in E$。因此 v 既不是 a 也不是 b，而且根据同样的理由可得出要么 $\{a, v\}\in E$，要么 $\{b, v\}\in E$。在其中任何一种情形里，这都给出在 G 里从 u 到 v 长度小于或等于 3 的通路，矛盾。

    b) 假设 $G=(V, E)$。设 $a, b\in V$。必须证明在 $\bar{G}$ 里 a 和 b 之间的距离不超过 3。若 $\{a, b\}\notin E$，则得出这个结果，所以假设 $\{a, b\}\in E$。因为 G 的直径大于或等于 3，所以存在顶点 u 和 v，使得 u 和 v 之间的距离在 G 里大于或等于 3。要么 u 要么 v，或者它们二者是不属于集合 $\{a, b\}$ 的。假设 u 是与 a 和 b 都不同的。要么 $\{a, u\}\in E$ 要么 $\{b, u\}\in E$；否则 a, u, b 是 $\bar{G}$ 里长度为 2 的通路。所以，不失一般性，假设 $\{a, u\}\in E$。因此 v 是与 a 和 b 都不同的。若 $\{a, v\}\in E$，则 u, a, v 是 G 里长度为 2 的通路，所以 $\{a, v\}\notin E$ 而且因此 $\{b, v\}\in E$（否则在 $\bar{G}$ 里有长度为 2 的通路 a, v, b）。因此 $\{u, b\}\notin E$；否则 u, b, v 是 G 里长度为 2 的通路。因此，a, v, u, b 是 $\bar{G}$ 里长度为 3 的通路，即为所求。

41. a, b, e, z

43. a, c, b, d, e, z

45. 若 G 是可平面的，则因为 $e\leqslant 3v-6$，所以 G 至多有 27 条边。（若 G 是不连通的，则它的边甚至更少。）同理，$\bar{G}$ 至多有 27 条边。但是 G 与 $\bar{G}$ 的并图是 K_{11}，它有 55 条边，而且 $55>27+27$。

47. 假设 G 是用 k 种颜色着色的并且独立数为 i。因为每个颜色类必须是一个独立集，所以每个颜色类只有不超过 i 个元素。因此至多存在 ki 个顶点。

49. a) 根据 4.5 节的定理 1，选择恰好 m 条边的概率是 $C(n, m)\ p^{m}\ (1-p)^{n-m}$。

    b) 根据 4.5 节的例 14，期望值是 np。

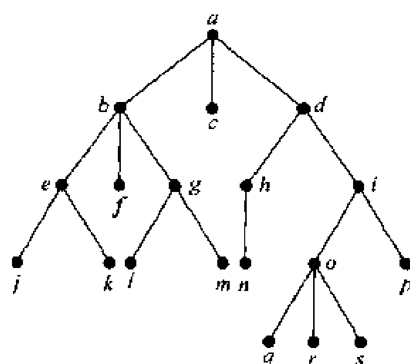
    c) 为了生成带标记的图 G，当把这个过程应用到成对的顶点时，所选择的随机数 x 必须是这样的：当 G 在那对顶点之间有边时，x 小于或等于 1/2；当 G 在那对顶点之间没有边时，x 大于 1/2。因此对每条边来说，作出正确选择的概率是 1/2；而对所有边来说，这个概率是 $1/2^{C(n,2)}$。因此所有带标记的图都是等可能的。

51. 假设 P 是单调增的。假如每当从简单图删除边时，不保持没有 P 这个性质。那么将存在没有 P 的简单图 G 和另一个有 P 的简单图 G'，G' 与 G 有相同的顶点，而 G' 的边是删除一些 G 的边之后剩下的边。但 P 是单调增的，所以由于 G' 有 P，因此通过给 G' 添加边而得到的 G 也有 P，矛盾。逆命题的证明是类似的。

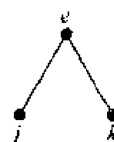
第 8 章

8.1 节

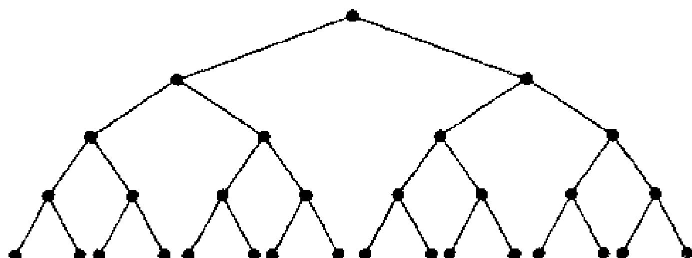
1. a)，c)，e)
3. 否
5. a)



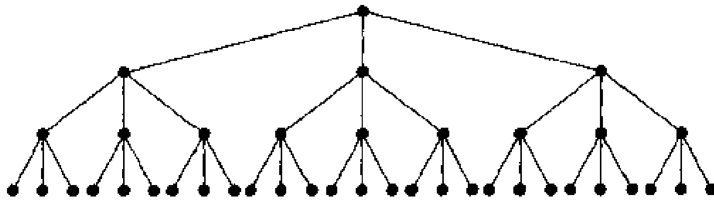
b) c c)



7. a) 2 b) 4 c) 9
9. “仅当”部分是定理 2 和树的定义。假设 G 是带 n 个顶点和 $n-1$ 条边的连通简单图。若 G 不是树，则根据练习 8，它包含这样一条边：删除这条边就产生一个图 G'，G'仍然是连通的。若 G'不是树，则删除一条边来产生连通图 G''。重复这个过程，直到结果是树为止。这需要至多 $n-1$ 步，因为只有 $n-1$ 条边。根据定理 2，结果的图有 $n-1$ 条边，因为它有 n 个顶点。所以不需要删除边，G 已经是树了。
11. 9 999
13. 2 000
15. 999
17. 1 000 000 美元
19. 根据定理 4，这样的树不存在，因为对 $m=2$ 或 $m=84$ 来说这是不可能的。
21. 高度为 4 的完全二叉树：



高度为 3 的完全三元树：



23. a) 根据定理 3，有 $n = mi + 1$。因为 $i + l = n$，所以有 $l = n - i$，所以 $l = (mi + 1) - i = (m - 1)i + 1$。

    b) 有 $n = mi + 1$ 和 $i + l = n$。因此 $i = n - l$。所以 $n = m(n - l) + 1$。对 n 求解就给出 $n = (ml - 1)(m - 1)$。从 $i = n - l$ 得出 $i = [(ml - 1)/(m - 1)] - l = (l - 1)/(m - 1)$。

25. $n - t$

27. a) 1　　b) 3　　c) 5

29. a) 父目录

    b) 子目录或包含的文件

    c) 在相同父目录里的子目录或包含的文件

    d) 在路径名称里的所有目录

    e) 在该目录或该目录的子目录里的所有子目录和包含的文件，依次类推。

    f) 到这个目录或文件的路径的长度

    g) 系统的深度，即最长路径的长度

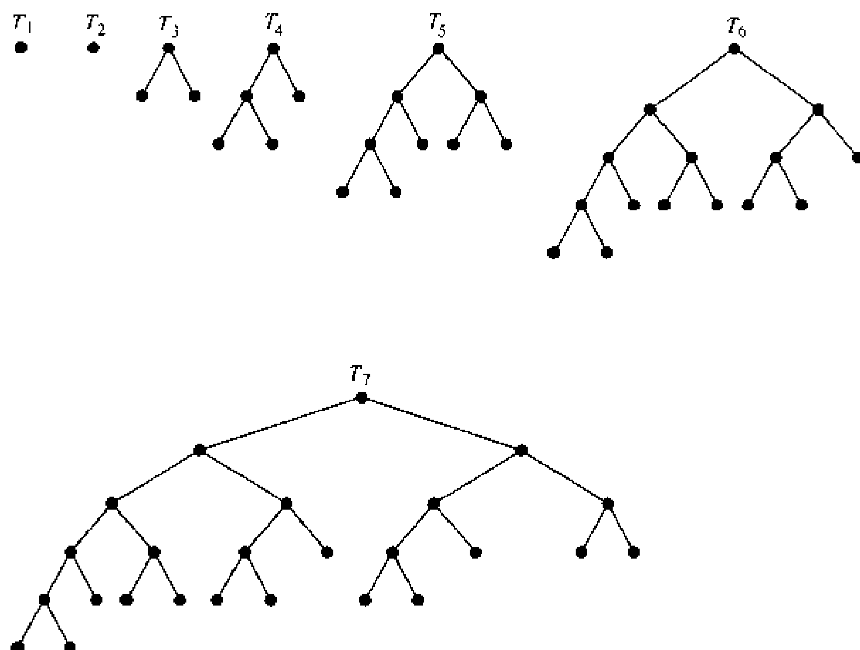
31. 设 $n = 2^k$，其中 k 是正整数。若 $k = 1$，则没有什么要证明的，因为可以用 $n - 1 = 1$ 个处理器在 $\log 2 = 1$ 步里把两个数相加。假定可以在 $\log n$ 步里用 $n - 1$ 个处理器的树型连接网络对 $n = 2^k$ 个数求和。设 $x_1, x_2, \cdots, x_{2n}$ 是希望求和的 $2n = 2^{k+1}$ 个数。$2n - 1$ 个处理器的树型连接网络包括 $n - 1$ 个处理器的树型连接网络，以及作为每个树叶儿子的两个新处理器。在一步之内，可以用这个较大的网络的树叶来求出 $x_1 + x_2, x_3 + x_4, \cdots, x_{2n-1} + x_{2n}$，结果得出 n 个数，根据归纳假设，可以用网络的其余部分在 $\log n$ 步内求出它们的和。因为使用了 $\log n + 1$ 步而且 $\log(2n) = \log 2 + \log n = 1 + \log n$，所以这样就完成了证明。

33. 只有 c

35. c 和 h

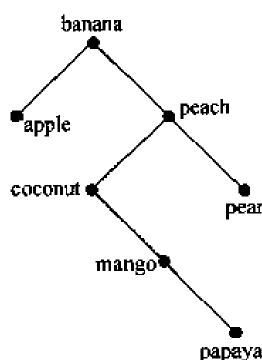
37. 假设树 T 有至少两个中心。设 u 和 v 是不同的中心，都有离心度 e，u 和 v 不相邻。因为 T 是连通的，所以有从 u 到 v 的简单通路 P。设 c 是这个通路上的任意顶点。因为 c 的离心度至少为 e，所以存在顶点 w 使得从 c 到 w 的唯一简单通路长度至少为 e。显然这条铁路不能同时包含 u 和 v，否则将有一条简单回路。事实上，一旦这条从 c 到 w 的通路可能沿着 P 的一部分向 u 或 v 前进，它就离开 P 并且不返回 P。不失一般性，假定这条通路不沿着 P 向 u 前进。于是从 u 到 c 到 w 的通路是简单的，而且长度不超过 e，矛盾。因此 u 和 v 是相邻的。现在因为任何两个中心都是相邻的，所以假如有多于两个的中心，那么 T 就包含简单回路 K_3 作为子图，矛盾。

39.



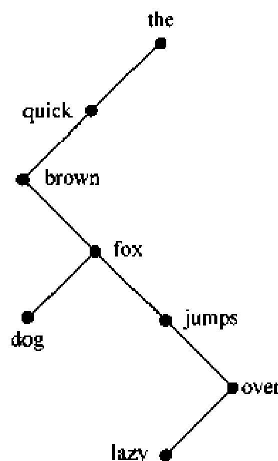
8.2 节

1.



3. a) 3　　b) 1　　c) 4　　d) 5

5.



7. 至少需要$\lceil \log_3 4 \rceil = 2$次称重，因为只有四种结果（因为不要求确定硬币是较轻还是较重）。事实上，两次称重是足够的。首先称重硬币 1 和硬币 2。若它们平衡，则称重硬币 1 和硬币 3。若硬币 1 与硬币 3 重量相同，则硬币 4 是伪币，若它们重量不相同，则硬币 3 是伪币。若硬币 1 与硬币 2 重量不相同，则再称重硬币 1 和硬币 3。若它们平衡，则硬币 2 是伪币；若它们不平衡，则硬币 1 是伪币。

9. 至少需要$\lceil \log_3 13 \rceil = 3$次称重。事实上，三次称重是足够的。首先把硬币 1，2 和 3 放在天平的左边，而把硬币 4，5 和 6 放在天平的右边。若相等，则应用例 2 到硬币 1，2，7，8，9，10，11 和 12 上。若不相等，则应用例 2 到硬币 1，2，3，4，5，6，7

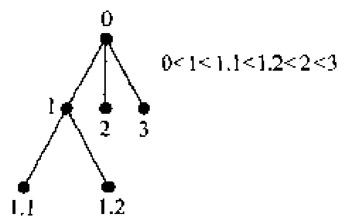
和 8 上。

11. a) 是　b) 否　c) 是　d) 是

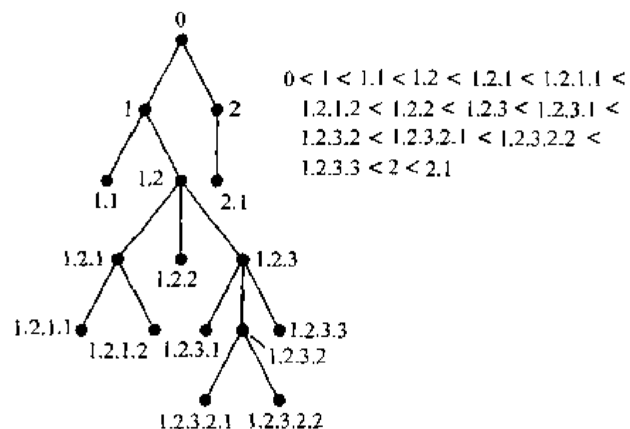
13. a: 000, e: 001, i: 01, k: 1100, o: 1101, p: 11110, u: 11111

8.3 节

1.



3.



5. 否

7. a, b, d, e, f, g, c

9. a, b, e, k, l, m, f, g, n, r, s, c, d, h, o, i, j, p, q

11. d, b, i, e, m, j, n, o, a, f, c, g, k, h, p, l

13. d, f, g, e, b, c, a

15. k, l, m, e, f, r, s, n, g, b, c, o, h, i, p, q, j, d, a

17. a) $- * \uparrow + x 2 3 - y + 3 x 5$

b) $x\ 2 + 3 \uparrow y 3 x + - * 5 -$

c) $((((x+2) \uparrow 3) * (y-(3+x))) - 5)$

19. a) $+ + x * xy / xy$, $+ x / + * xyxy$

b) $xxy * + xy / +$, $xxy * x + y / +$

c) $((x+(x*y))+(x/y))$, $(x+(((x*y)+x)/y))$

21. a) $\leftrightarrow \neg \wedge pq \vee \neg p \neg q$, $\vee \wedge \neg p \leftrightarrow q \neg p \neg q$

b) $pq \wedge \neg p \neg q \neg \vee \leftrightarrow$, $p \neg qp \neg \leftrightarrow \wedge q \neg \vee$

c) $(((p \wedge q) \neg) \leftrightarrow ((p \neg) \vee (q \neg)))$, $(((p \neg) \wedge (q \leftrightarrow (p \neg))) \vee (q \neg))$（一元运算是在它们的运算对象之后）

23. a) $- \cap AB \cup A - BA$

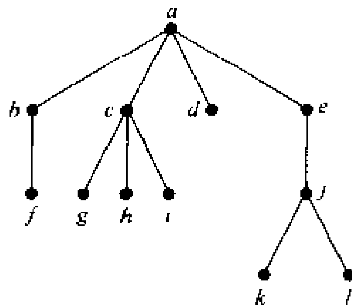
b) $AB \cap ABA - \cup -$

c) $((A \cap B) - (A \cup (B - A)))$

25. 14

27. a) 1　b) 1　c) 4　d) 2205

29.



31. 用数学归纳法。对一个元素的表来说结果是平凡的。假定对 n 个元素的表来说结果为真。对于归纳步骤，从后面开始。找出表后面的顶点序列，从最后一个树叶开始，到根结束，每个顶点都是它后面那个顶点的最后一个儿子。删除这个树叶并且应用归纳假设。

33. 在每种情形里分别为 c，d，b，f，g，h，e，a。

35. 用数学归纳法证明。设 $S(X)$ 和 $O(X)$ 分别表示合式公式 X 里的符号数和运算数。对长度为 1 的合式公式来说命题为真，因为它们都有 1 个符号和 0 个运算。假定对长度小于 n 的合式公式来说命题为真。长度为 n 的合式公式必然形如 $*XY$，其中 $*$ 是运算而 X 和 Y 都是长度小于 n 的合式公式。于是根据归纳假设 $S(*XY)=S(X)+S(Y)=(O(X)+1)+(O(Y)+1)=O(X)+O(Y)+2$。因为 $O(*XY)=1+O(X)+O(Y)$，所以 $S(*XY)=O(*XY)+1$。

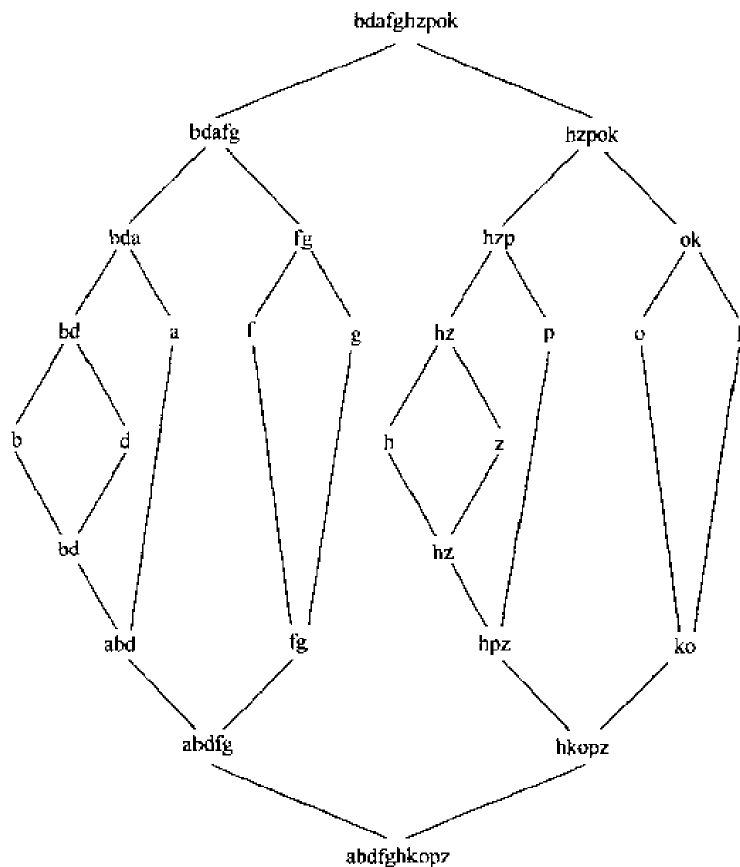
37. 例如，$xy+zx\circ+x\circ$，$xyz++yx++$，$xyxy\circ\circ xy\circ\circ z\circ+$，$xz\times zz+\circ$，$yyyy\circ\circ\circ$，$zx+yz+\circ$

8.4 节

1. 在第 1 遍结尾：1，3，5，4，7；在第 2 遍结尾：1，3，4，5，7；在第 3 遍结尾：1，3，4，5，7；在第 4 遍结尾：1，3，4，5，7

3.
```
procedure better bubblesort (a1, …, an: 整数)
i := 1; done := false
while (i < n 和 done = false)
begin
   done := true
   for j := 1 to n - i
    if aj > aj+1 then
    begin
      交换 aj 和 aj+1
      done := false
    end
   i := i + 1
end {a1, …, an 排列成升序}
```

5.



7. 设两个表分别是$1,2,\cdots,m-1,m+n-1$ 和 $m,m+1,\cdots,m+n-2,m+n$。

9. a) 1, 5, 4, 3, 2; 1, 2, 4, 3, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5

b) 1, 4, 3, 2, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5

c) 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5

11. $O(n^2)$

13. $n-1$

15. 6

17. 在最坏情形下$O(n^2)$

19. **procedure** *mergesort* (a_1, $\cdots$, a_n: 整数)

$m := \lceil n/2 \rceil$

if $n > 1$ **then**

begin

$L_1 := (a_1, \cdots, a_m)$

$L_2 := (a_{m+1}, \cdots, a_n)$

$L_1 :=$ *mergesort* (L_1); $L_2 :=$ *mergesort* (L_2)

$L :=$ *merge* (L_1, L_2)

end

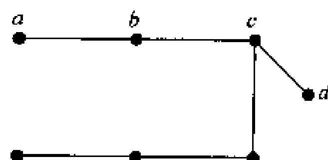
else $L := (a_1)$ {只带有一个元素的表是排序的}

{L 是排序的}

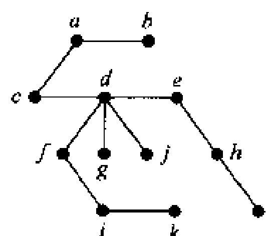
8.5 节

1. $m-n+1$

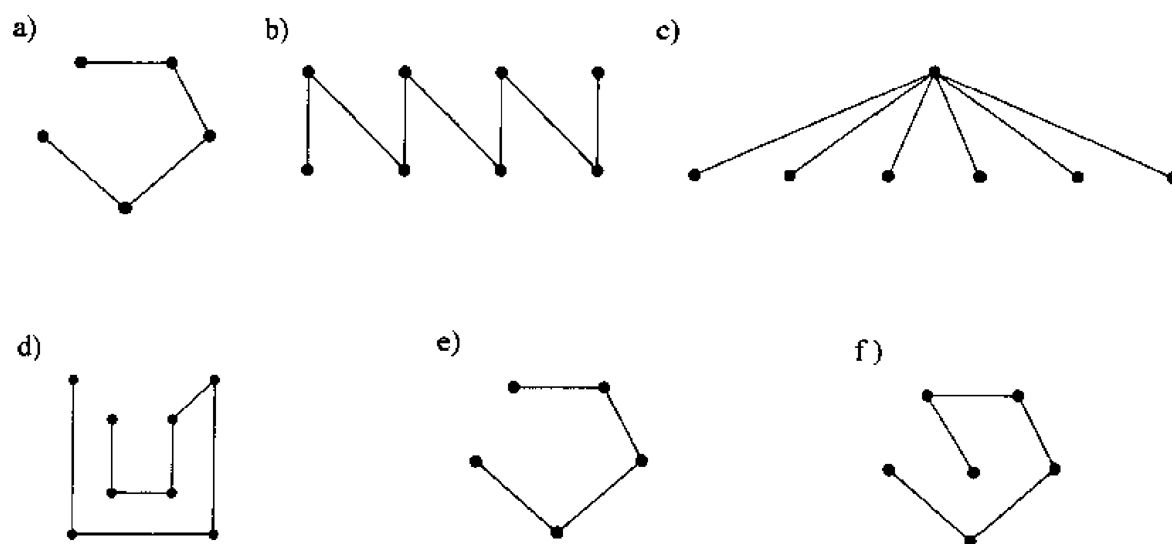
3.



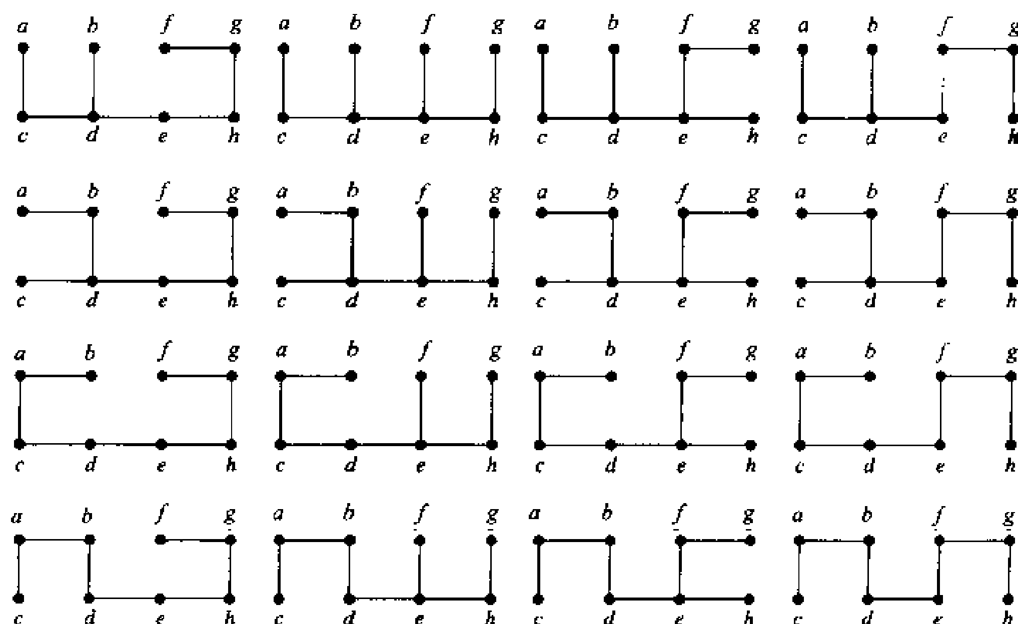
5.



7.



9.



11. a) 3 b) 16 c) 4 d) 5

13.

15.

17. 一组可能不再继续下去的航班是：波士顿－纽约，底特律－波士顿，波士顿－华盛顿，纽约－华盛顿，纽约－芝加哥，亚特兰大－华盛顿，亚特兰大－达拉斯，亚特兰大－洛杉矶，亚特兰大－圣路易斯，圣路易斯－达拉斯，圣路易斯－底特律，圣路易斯－丹佛，达拉斯－圣迭戈，达拉斯－洛杉矶，达拉斯－旧金山，圣迭戈－洛杉矶，洛杉矶－旧金山，旧金山－西雅图。

19. 树

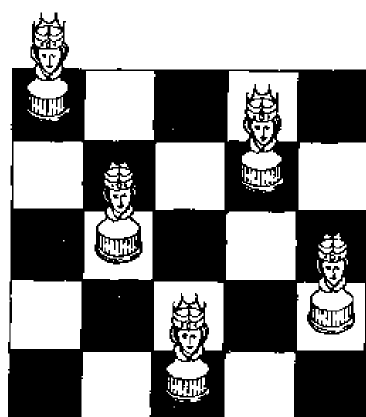
21. **procedure** *depth first search*（G：带有有序顶点 v_1，…，v_n 的简单图）

```
T := 以 v1 作为根且没有其他顶点的根树
visit (v1)
{T 是所需要的树}
procedure visit (v)
for v 的每个邻居 w
begin
  if w 不在 T 里 then
  begin
    把顶点 w 和边{v,w}放到 T 里
    visit (w)
  end
end
```

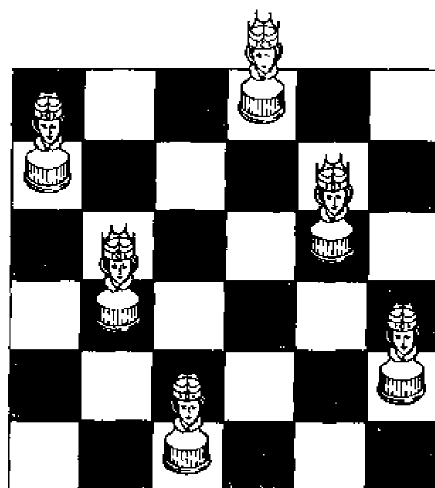
23. 通过对通路的长度进行归纳来证明：若通路长度为 0，则结果是平凡的。若长度为 1，则 u 是与 v 相邻的，所以 u 是在宽度优先生成树的 1 层上。假定对长度为 l 的通路来说结果为真。若通路长度为 $l+1$，则设 u' 是从 v 到 u 的最短通路上倒数第二个顶点。根据归纳假设，u' 是在宽度优先生成树的 1 层上。假如 u 是在不超过 l 的层上，那么显然从 v 到 u 的最短通路的长度也不超过 l。所以在添加 l 层的顶点之后，仍然没有添加 u 到宽度优先生成树上。因为 u 是与 u' 相邻的，所以将把它添加在 $l+1$ 层上（尽管连接 u' 和 u 的边不必添加）。

25. a) 无解

b)

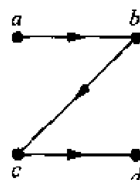


c)

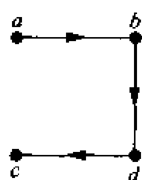


27. 从某个顶点开始并且沿着一条通路前进，尽量不重复经过顶点，在所有顶点都已经访问过之后，允许返回到出发点。当不可能沿着一条通路继续下去时，就回溯并且尝试当前通路的另外一种扩充。
29. 取 G 的连通分支的生成树的并图。它们都是不相交的，所以结果是一个森林。
31. $m-n+c$
33. 在每个分支上使用深度优先搜索。
35. 设 T 是在图 8-39 里构造的生成树，而 T_1, T_2, T_3 和 T_4 是在图 8-40 里的生成树。用 $d(T', T'')$ 表示 T' 与 T'' 之间的距离。则 $d(T,T_1)=6, d(T,T_2)=4, d(T,T_3)=4, d(T,T_4)=2, d(T_1,T_2)=4, d(T_1,T_3)=4, d(T_1,T_4)=6, d(T_2,T_3)=4, d(T_2,T_4)=2, d(T_3,T_4)=4$。
37. 假定 $e_1=\{u,v\}$ 是像规定的那样，则 $T_2\cup\{e_1\}$ 包含一个包含 e_1 的简单回路 C。图 $T_1-\{e_1\}$ 包含两个连通分支；e_1 的端点是在两个不同的分支里。从 u 开始按照与 e_1 相反的方向前进，直到到达 v 所在的分支里第一个顶点为止。刚刚经过的边是 e_2。显然 $T_2\cup\{e_1\}-\{e_2\}$ 是树，因为 e_2 是在 C 上。另外 $T_1-\{e_1\}\cup\{e_2\}$ 是树，因为 e_2 重新连接这两个分支。

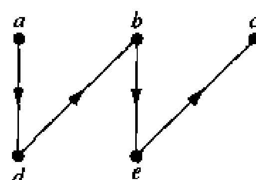
39. 练习24:



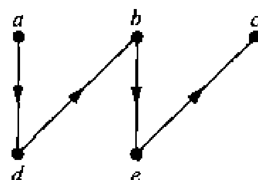
练习25:



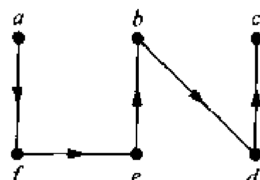
练习26:



练习27:



练习28:



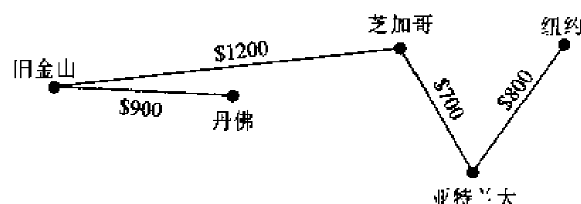
41. 首先构造这个有向图里的欧拉回路，然后从这个回路删除通向从前访问过的顶点的每条边。

8.6 节

1. 深泉镇－奥席斯，奥席斯－戴尔，奥席斯－舍尔弗皮克，舍尔弗皮克－哥德菲尔，利达－哥德波因特，哥德波因特－比提，利达－哥德菲尔，哥德菲尔－托诺帕，托诺帕－曼哈坦，托诺帕－温泉镇

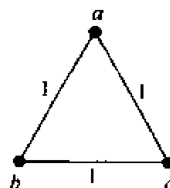
3. $\{e,f\},\{c,f\},\{e,h\},\{h,i\},\{b,c\},\{b,d\},\{a,d\},\{g,h\}$

5.



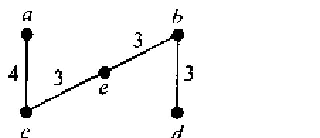
7. $\{e,f\},\{a,d\},\{h,i\},\{b,d\},\{c,f\},\{e,h\},\{b,c\},\{g,h\}$

9.

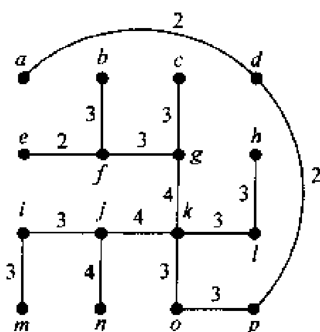


11. 不在每个阶段上选择权最小的边，在每个阶段上选择带同样性质的权最大的边。

13.



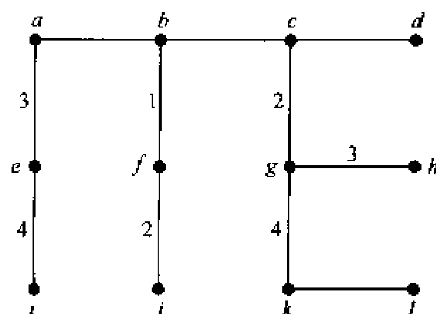
15.



17. 首先找出带有 n 条边的图 G 的最小生成树 T。然后对 $i=1$ 到 $n-1$，只从 G 删除 T 的第 i 条边并且找出剩下图的生成树。挑选出这 $n-1$ 个树中长度最短的一个。

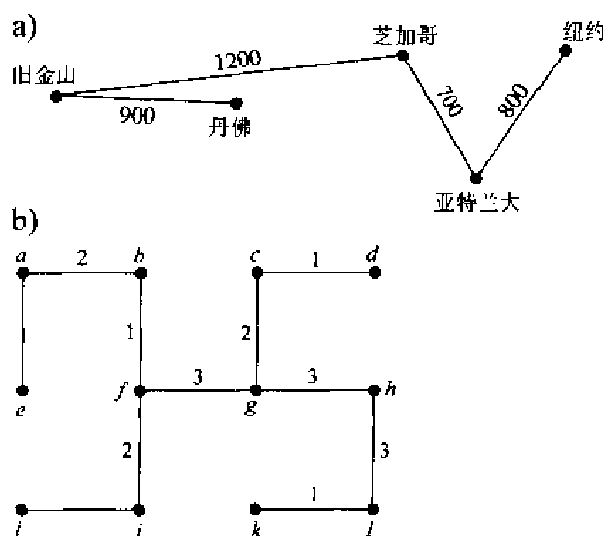
19. 若所有的边都有不同的权，则当添加一条边 e_{k+1} 到 T 并且删除一条边 e 时，代替可能产生另外一个生成树的是在普林算法的正确性证明里获得一个矛盾。

21.



23. 与克鲁斯卡算法一样，不同之处在于开始时 T：＝这组边，并且从 $i=1$ 到 $n-1-s$ 进行迭代，其中 s 是开始时的边数。

25.



27. 根据练习 24，在索林算法的每个阶段上都得出一个森林。因此在选择了 $n-1$ 条边之后，就得出一个树。剩下证明这个树是最小生成树。设 T 是与索林树 S 有着尽可能多的公共边的最小生成树。若 $T \neq S$，则存在边 $e \in S-T$，它是在算法的某个阶段上添加的，其中在那个阶段之前 S 里的所有边也都是在 T 里的。$T \cup \{e\}$ 包含唯一一个简单回路。在这个回路上找出边 $e' \in S-T$ 和边 $e'' \in T-S$，使得当把这个阶段的树看作是"超级顶点"时，e' 和 e'' 是"相邻的"。于是根据这个算法，就有 $w(e') \leqslant w(e'')$。所以用 $T-\{e'\} \cup \{e''\}$ 来替换 T 就产生了比 T 更接近 S 的最小生成树。

29. 这 r 个树中的每个都用一条新边连接到至少一个其他的树上。因此在结果里至多有 $r/2$ 个树（每个新树都包含两个或更多的旧树）。为了达到这个目的，需要添加 $r-(r/2)=r/2$ 条边。因为添加的边数是整数，所以它至少是 $\lceil r/2 \rceil$。

31. 若 $k \geqslant \log n$，则 $n/2^k \leqslant 1$，所以 $\lceil n/2^k \rceil = 1$，所以根据练习 30，这个算法在至多 $\log n$ 次迭代之后就结束。

补充练习

1. 假定 T 是树。那么显然 T 没有简单回路。若添加一条连接两个相邻顶点 u 和 v 的边 e，则显然形成一条简单回路，因为当添加 e 到 T 时所得出的图有太多的边而不能是树。所形成的唯一的简单回路是由边 e 以及从 v 到 u 的 T 中的唯一通路 p 来组成的。假定 T 满足所给定的条件。所需要做的全部事情就是证明 T 是连通的，因为在图中没有简单回路。假定 T 不是连通的。那么设 u 和 v 是在不同的连同分支里。添加 $e=\{u,v\}$ 就不满足这些条件。

3. 假定树 T 具有度分别为 $d_1, d_2, \cdots, d_n$ 的 n 个顶点。因为 $2e=\sum_{i=1}^{n} d_i$ 和 $e=n-1$，所以有 $2(n-1)=\sum_{i=1}^{n} d_i$。因为每个 $d_i \geqslant 1$，所以 $2(n-1)=n+\sum_{i=1}^{n}(d_i-1)$，或 $n-2=\sum_{i=1}^{n}(d_i-1)$。因此在这个和里至多有 $n-2$ 项可以是 1 或更大。因此，它们中至少两个是 0。所以对至少两个 i 值来说有 $d_i=1$。

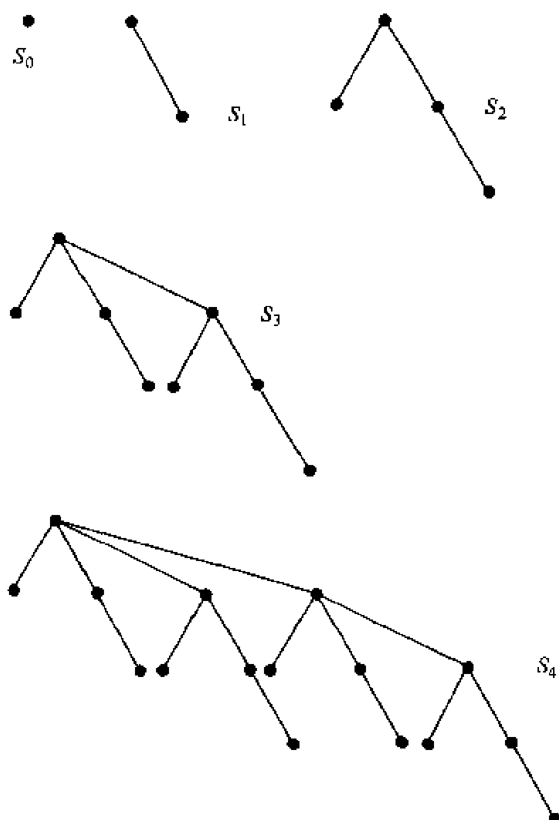
5. $2n-2$

7. T 没有回路，所以它不能有同胚于 $K_{3,3}$ 或 K_5 的子图。

9. 分别着色每个连通分支。对这些连通分支的每个来说，首先选择树根，然后着色所有偶数层上的顶点成红色并且着色所有奇数层上的顶点成蓝色。

11. 上界：k^h；下界：$2\lceil k/2 \rceil^{h-1}$

13.



15. 用数学归纳法。对 $k=0$ 来说结果是平凡的。假定对 $k-1$ 来说结果为真。T_{k-1}是 T 的父亲树。根据归纳法，T 的儿子树可以从 T_0，…，T_{k-2}以所述方式来获得。r_{k-2}到 r_{k-1}的最后连接正如在 S_k 树的定义中所述的那样。

17. **procedure** *level*（T：带根 r 的有序根树）

```
queue :=只包含根 r 的序列
while queue 包含至少一项
begin
  v :=队列里第一个顶点
  列出 v
  从队列里删除 v 并且把 v 的儿子都放到队尾
end
```

19. 这样建立树：为地址 0 插入根，然后为每个标记 i 的顶点插入一个子树，i 是正整数，这些子树是从每个标记为 $i.j$ 的顶点所对应的子树来建立的，j 是正整数，依次类推。

21. **procedure** *insertion*（a_1，…，a_n：实数）

```
for j := 2 to n
begin
  i := 1
  while a_j > a_i
    i := i + 1
```

$m := a_j$

for $k := 0$ **to** $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

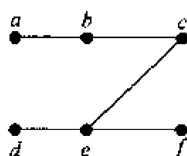
end $\{a_1, \cdots, a_n$ 已经排序$\}$

23. 若 u 是悬挂点并且 $e = \{u, v\}$ 是图中关联 u 的边，则图中没有关联 u 的其他边。所以 e 必然属于任何一个生成树，因为假如它不属于一个生成树，那么这个生成树就不包含关联 u 的边。

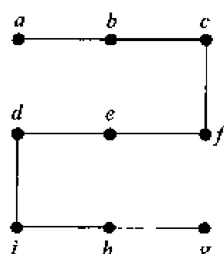
25. a) 是　　b) 否　　c) 是

27. 所得出的图没有边在多于一个所描述类型的简单回路上。因此它是仙人掌图。

29.

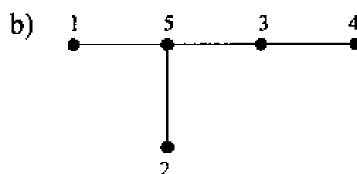


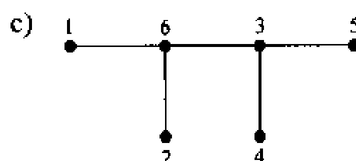
31.

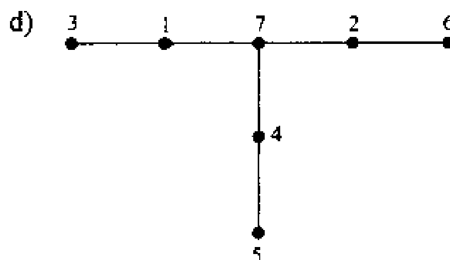


33.

a) 

b) 

c) 

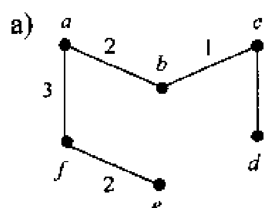
d) 

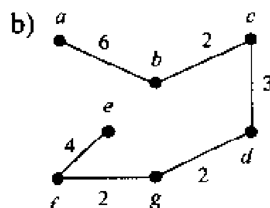
35. 6

37. a) 是　　b) 否　　c) 是

39. 设 G' 是从 G 删除顶点 v 和关联 v 的所有边而获得的图。可以这样获得 G 的最小生成树：选出关联 v 的权最小的边以及 G' 的最小生成树。

41.

a) 

b) 

第 9 章

9.1 节

1. a) 1　　b) 1　　c) 0　　d) 0

3. (0, 0) 和 (1, 1)

5. $x + xy = x \cdot 1 + xy = x(1 + y) = x(y + 1) = x \cdot 1 = x$

7.

x	y	z	xy	$y\bar{z}$	$\bar{x}z$	$x\bar{y} + y\bar{z} + \bar{x}z$	$\bar{x}y$	$\bar{y}z$	$x\bar{z}$	$\bar{x}y + \bar{y}z + x\bar{z}$
1	1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1	0	1
1	0	0	1	0	0	1	0	0	1	1
0	1	1	0	0	1	1	1	0	0	1
0	1	0	0	1	0	1	1	0	0	1
0	0	1	0	0	1	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0

9.

x	$x + x$	$x \cdot x$
0	0	0
1	1	1

11.

x	$x + 1$	$x \cdot 0$
0	1	0
1	1	0

13.

x	y	z	$y + z$	$x + (y + z)$	$x + y$	$(x + y) + z$	yz	$x(yz)$	xy	$(xy)z$
1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	0	1	0
1	0	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0
0	1	0	1	1	1	1	0	0	0	0
0	0	1	1	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

15.

x	y	xy	$(\overline{xy})$	$\bar{x}$	$\bar{y}$	$\bar{x}+\bar{y}$	$x+y$	$(\overline{x+y})$	$\bar{x}\bar{y}$
1	1	1	0	0	0	0	1	0	0
1	0	0	1	0	1	1	1	0	0
0	1	0	1	1	0	1	1	0	0
0	0	0	1	1	1	1	0	1	1

17.

x	y	$x\oplus y$	$x+y$	xy	$(\overline{xy})$	$(x+y)(\overline{xy})$	$x\bar{y}$	$\bar{x}y$	$x\bar{y}+\bar{x}y$
1	1	0	1	1	0	0	0	0	0
1	0	1	1	0	1	1	1	0	1
0	1	1	1	0	1	1	0	1	1
0	0	0	0	0	1	0	0	0	0

19. a) 对，用真值表可证。

b) 错，如取 $x=1$，$y=1$，$z=1$。

c) 错，如取 $x=1$，$y=1$，$z=0$。

21. 根据德摩根律，对一个表达式的求补，除了变元取补外，其余如同求此表达式的对偶。

23. 16

25. 由控制律、分配律和同一律，$x\vee x=(x\vee x)\wedge 1=(x\vee x)\wedge(x\vee\bar{x})=x\vee(x\wedge\bar{x})=x\vee 0=x$。类似地，$x\wedge x=(x\wedge x)\vee 0=(x\wedge x)\vee(x\wedge\bar{x})=x\wedge(x\vee\bar{x})=x\wedge 1=x$。

27. 由同一律和交换律，$0\vee 1=1$ 且 $0\wedge 1=0$，从而 $\bar{0}=1$。类似地，由于 $1\vee 0=1$ 且 $1\wedge 0=0$，从而 $\bar{1}=0$。

29. 首先注意 $x\wedge 0=0$ 且 $x\vee 1=1$ 对任意 x 成立，这很容易证明。为证明第一个等式，只要证明 $(x\vee y)\vee(\bar{x}\wedge\bar{y})=1$ 且 $(x\vee y)\wedge(\bar{x}\wedge\bar{y})=1$。由结合律、交换律、分配律、控制律和同一律知：$(x\vee y)\vee(\bar{x}\wedge\bar{y})=y\vee(x\vee(\bar{x}\wedge\bar{y}))=y\vee((x\vee\bar{x})\wedge(x\vee\bar{y}))=y\vee(1\wedge(x\vee\bar{y}))=y\vee(x\vee\bar{y})=(y\vee\bar{y})\vee x=1\vee x=1$，且 $(x\vee y)\wedge(\bar{x}\wedge\bar{y})=\bar{y}\wedge(\bar{x}\wedge(x\vee y))=\bar{y}\wedge((\bar{x}\wedge x)\vee(\bar{x}\wedge y))=\bar{y}\wedge(0\vee(\bar{x}\wedge y))=\bar{y}\wedge(\bar{x}\wedge y)=\bar{x}\wedge(y\wedge\bar{y})=\bar{x}\wedge 0=0$。类似可证第二个等式。

31. 由假设、练习 25 和分配律可得：$x=x\vee 0=x\vee(x\vee y)=(x\vee x)\vee y=x\vee y=0$。类似有 $y=0$。为证明第二个等式，注意 $x=x\wedge 1=x\wedge(x\wedge y)=(x\wedge x)\wedge y=x\wedge y=1$。类似有 $y=1$。

33. 用第 6 章补充练习 39 和 41 以及有补分配格的定义来建立定义中的五对定律。

9.2 节

1. a) $\bar{x}\bar{y}z$　　b) $\bar{x}y\bar{z}$　　c) $\bar{x}yz$　　c) $\bar{x}\bar{y}\bar{z}$

3. a) $xyz+xy\bar{z}+x\bar{y}z+x\bar{y}\bar{z}+\bar{x}yz+\bar{x}y\bar{z}+\bar{x}\bar{y}z$

b) $xyz+xy\bar{z}+\bar{x}yz$

c) $xyz+xy\bar{z}+x\bar{y}z+x\bar{y}\bar{z}$

d) $x\bar{y}z+x\bar{y}\bar{z}$

5. $wxy\bar{z} + wxyz + w\bar{x}yz + \bar{w}xyz + \bar{w}x\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}y\bar{z} + w\bar{x}\bar{y}\bar{z}$

7. a) $\bar{x} + \bar{y} + z$

   b) $x + y + z$

   c) $x + \bar{y} + z$

9. $y_1 + y_2 + \cdots + y_n = 0$ 当且仅当 $y_i = 0$ 对 $i = 1, 2, \cdots, n$ 都成立。本题成立当且仅当：若 $y_i = x_i$，则 $x_i = 0$；若 $y_i = \bar{x}_i$，则 $x_i = 1$。

11. a) $x + y + z$

   b) $(x + y + z)(x + y + \bar{z})(x + \bar{y} + z)(\bar{x} + y + z)(\bar{x} + y + \bar{z})$

   c) $(x + y + z)(x + y + \bar{z})(x + \bar{y} + z)(x + \bar{y} + \bar{z})$

   d) $(x + y + z)(x + y + \bar{z})(x + \bar{y} + z)(x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{x} + \bar{y} + \bar{z})$

13. a) $x + y + z$　　b) $x + \overline{(y + \overline{(\bar{x} + z)})}$

   b) $\overline{(x + \bar{y})}$　　c) $\overline{(x + \overline{(x + \bar{y} + z)})}$

15.

a)

x	$\bar{x}$	$x \downarrow x$
1	0	0
0	1	1

b)

x	y	xy	$x \downarrow x$	$y \downarrow y$	$(x \downarrow x) \downarrow (y \downarrow y)$
1	1	1	0	0	1
1	0	0	0	1	0
0	1	0	1	0	0
0	0	0	1	1	0

c)

x	y	$x + y$	$(x \downarrow y)$	$(x \downarrow y) \downarrow (x \downarrow y)$
1	1	1	0	1
1	0	1	0	1
0	1	1	0	1
0	0	0	1	0

17. a) $(((x \mid x) \mid (y \mid y)) \mid ((x \mid x) \mid (y \mid y))) \mid (z \mid z)$

   b) $(((x \mid x) \mid (z \mid z)) \mid y) \mid (((x \mid x) \mid (z \mid z)) \mid y)$

   c) x

   d) $(x \mid (y \mid y)) \mid (x \mid (y \mid y))$

19. 用 + 和 · 不能表示 $\bar{x}$，因为当输入为 1 时，无法取得值 0。

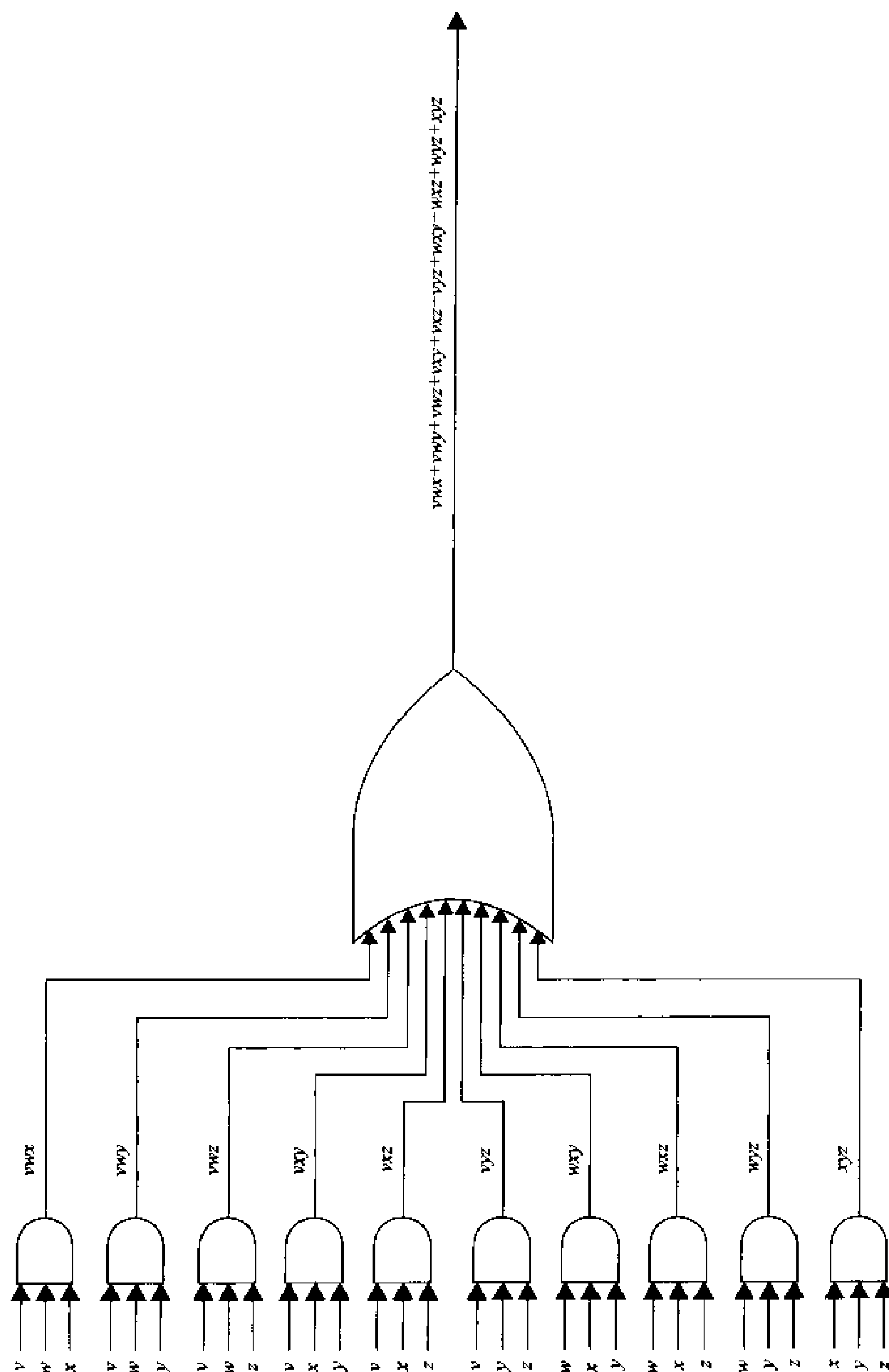
9.3 节

1. $(x + y)\bar{y}$

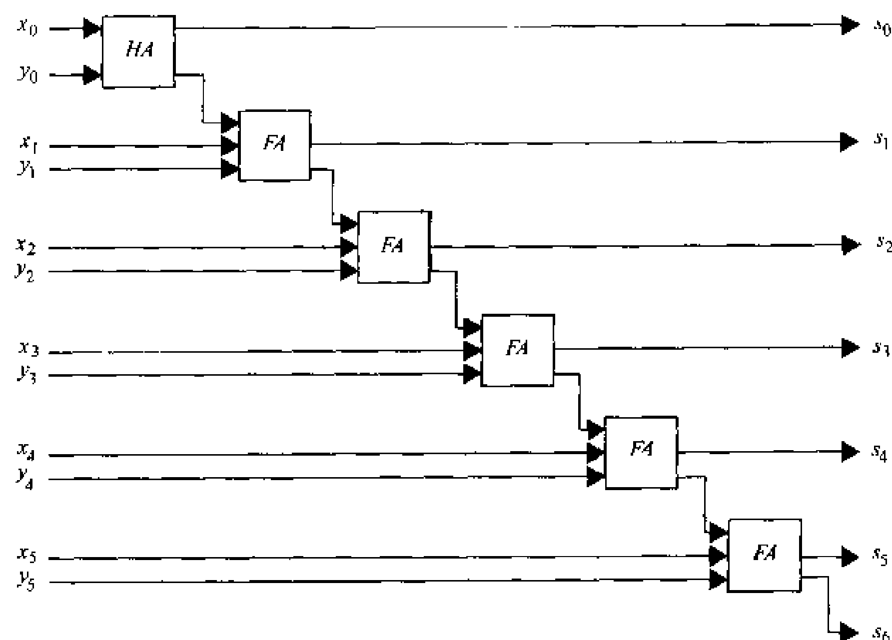
3. $\overline{(xy)} + (\bar{z} + x)$

5. $(x + y + z) + (\bar{x} + y + z) + (\bar{x} + \bar{y} + \bar{z})$

7.


vwx+vwy+vwz+vxy+vxz+vyz+wxy+wxz+wyz+xyz
vwx
vwy
vwz
vxy
vxz
vyz
wxy
wxz
wyz
xyz

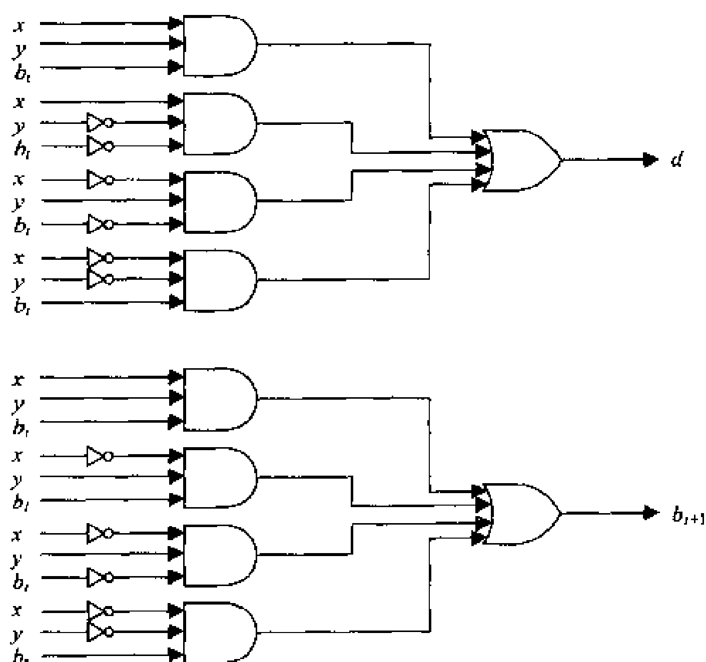
9.



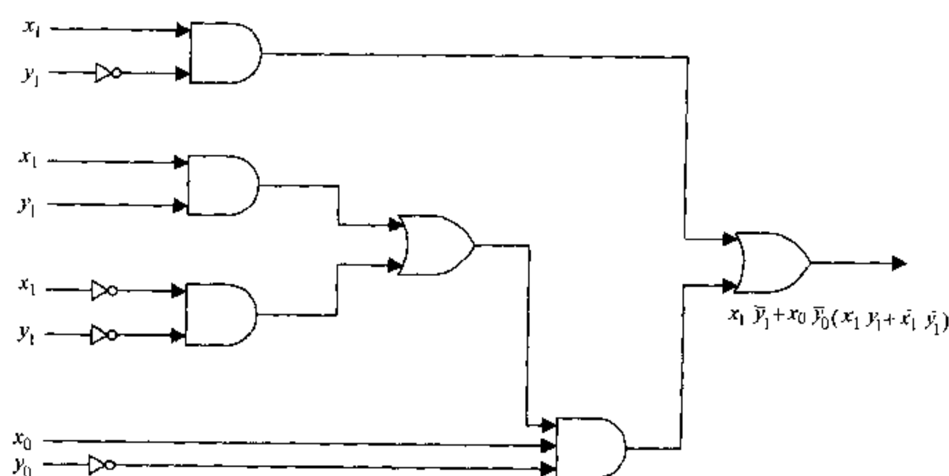
HA – 半加器

FA = 全加器

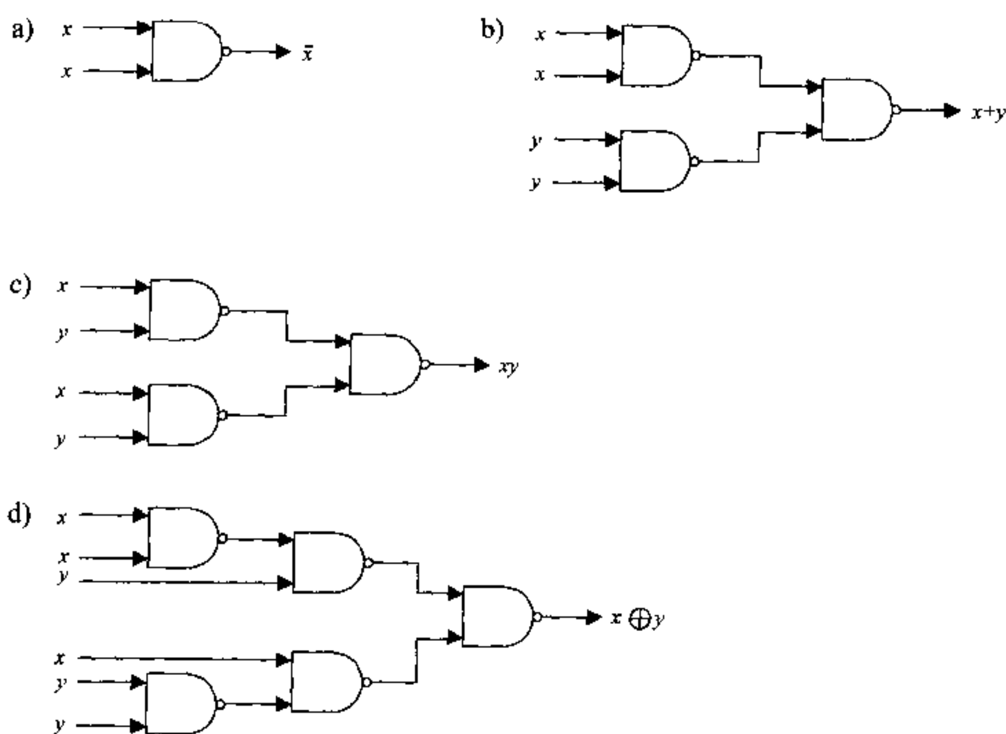
11.



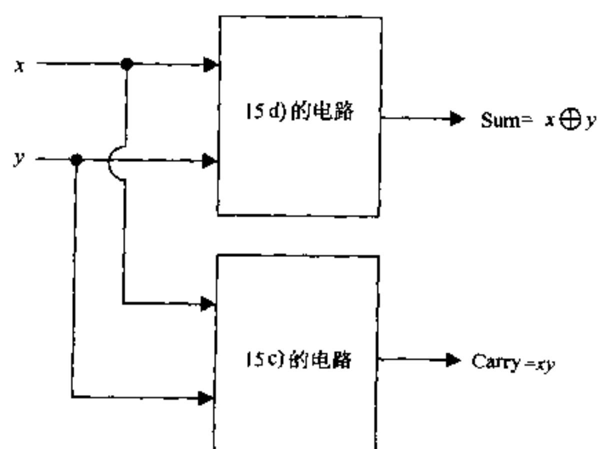
13.


x_1
y_1
x_1
y_1
x_1
y_1
x_0
y_0
$x_1\bar{y}_1+x_0\bar{y}_0(x_1y_1+\bar{x}_1\bar{y}_1)$

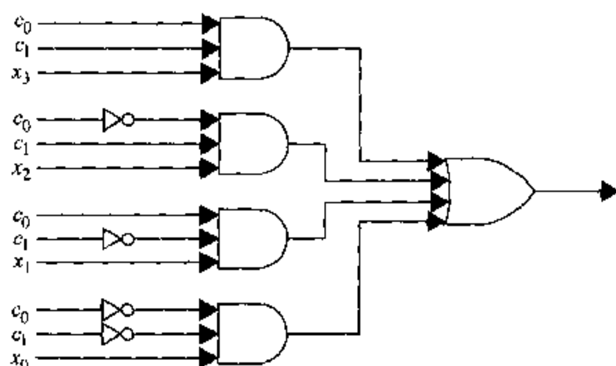
15.


a)
b)
c)
d)
$\bar{x}$
$x+y$
xy
$x\oplus y$

17.


15 d)的电路
15 c)的电路
Sum= $x\oplus y$
Carry $=xy$

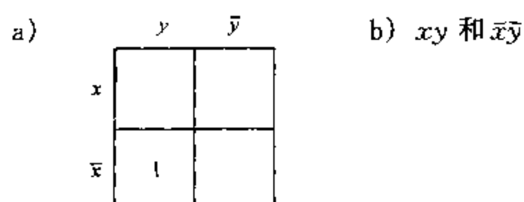
19.



9.4 节

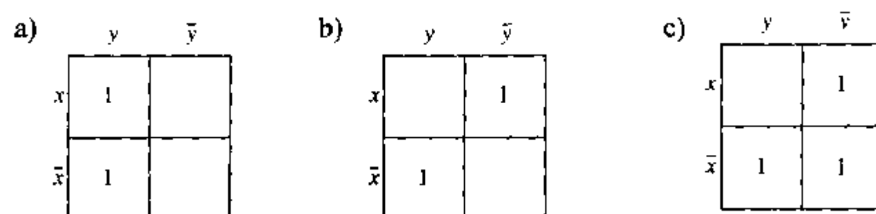
1.

a)



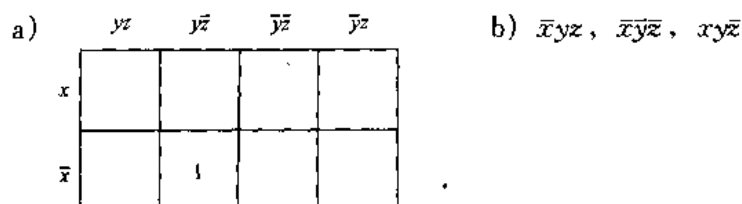
b) xy 和 $\bar{x}\bar{y}$

3.



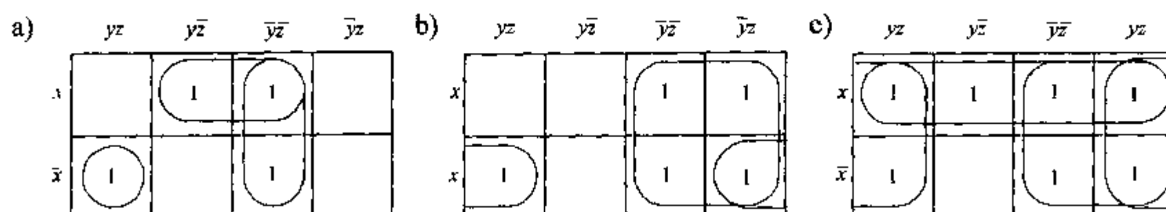
5.

a)



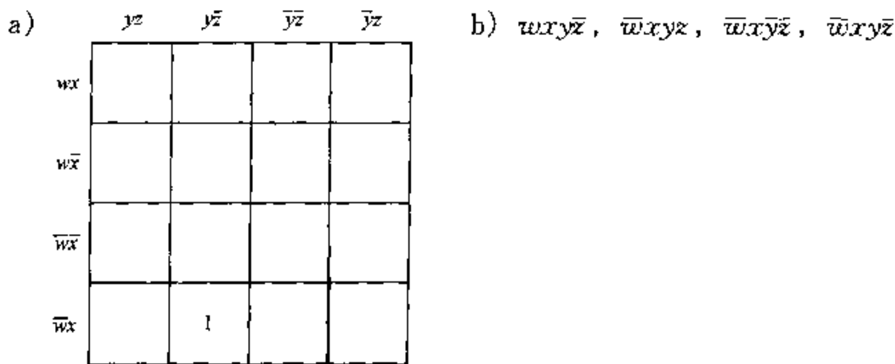
b) $\bar{x}yz$，$\bar{x}\bar{y}\bar{z}$，$xy\bar{z}$

7.



9.

a)

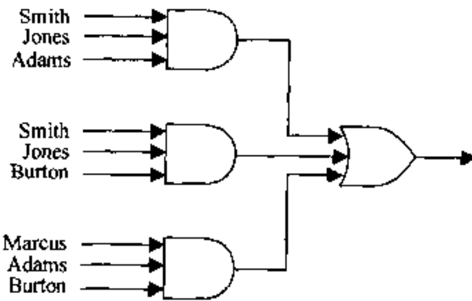


b) $wxy\bar{z}$, $\bar{w}xyz$, $\bar{w}x\bar{y}\bar{z}$, $\bar{w}xy\bar{z}$

11. a) 32

b) 5

13.



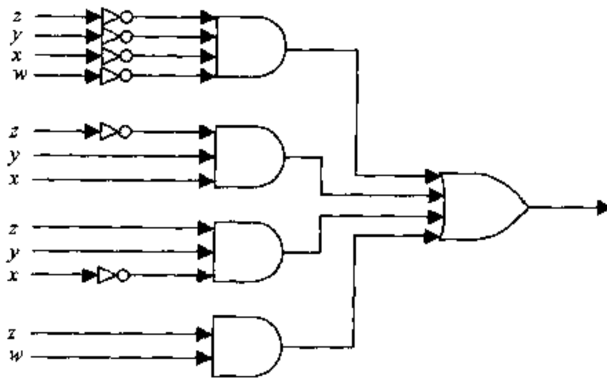
15. a) $\bar{x}z$ b) y c) $x\bar{z} + \bar{x}z + \bar{y}z$ d) $xz + \bar{x}y + y\bar{z}$

17. a) $wxz + wx\bar{y} + w\bar{y}z + w\bar{x}y\bar{z}$ b) $x\bar{y}z + \bar{w}\bar{y}z + wxy\bar{z} + w\bar{x}yz + \bar{w}\bar{x}y\bar{z}$

c) $\bar{y}z + wxz + w\bar{x}\bar{y} + \bar{w}\bar{x}y\bar{z}$ d) $wy + yz + \bar{x}y + wxz + \bar{w}\bar{x}z$

19. $x(y+z)$

21.



23. $\bar{x}\bar{z} + xz$

补充练习

1. a) $x=0$, $y=0$, $z=0$; $x=1$, $y=1$, $z=1$。

b) $x=0$, $y=0$, $z=0$; $x=0$, $y=0$, $z=1$; $x=0$, $y=1$, $z=0$; $x=1$, $y=0$, $z=1$; $x=1$, $y=1$, $z=0$; $x=1$, $y=1$, $z=1$。

c) 无值。

3. a) 是　　b) 不是　　c) 不是　　d) 是

5. $2^{2^{n-1}}$。

7. a) 若$F(x_1, \ldots, x_n)=1$，由控制律得：$(F+G)(x_1, \ldots, x_n)=F(x_1, \ldots, x_n)+G(x_1, \ldots, x_n)=1$。故 $F \leqslant F+G$。

 b) 若 $(FG)(x_1, \ldots, x_n)=1$，则 $F(x_1, \ldots, x_n) \cdot G(x_1, \ldots, x_n)=1$，故 $F(x_1, \ldots, x_n)=1$。由此得 $FG \leqslant F$。

9. 因为$F(x_1, \ldots, x_n)=1$ 蕴涵$F(x_1, \ldots, x_n)=1$，故$\leqslant$是自反的。设 $F \leqslant G$ 且 $G \leqslant F$，则$F(x_1, \ldots, x_n)=1$ 当且仅当 $G(x_1, \ldots, x_n)=1$，这蕴涵 $F=G$，故$\leqslant$是反对称的。设 $F \leqslant G \leqslant H$，则由$F(x_1, \ldots, x_n)=1$ 可得 $G(x_1, \ldots, x_n)=1$，此蕴涵$H(x_1, \ldots, x_n)=1$，从而 $F \leqslant H$，故$\leqslant$是传递的。

11. a) $x=1$，$y=0$，$z=0$

 b) $x=1$，$y=0$，$z=0$

 c) $x=1$，$y=0$，$z=0$

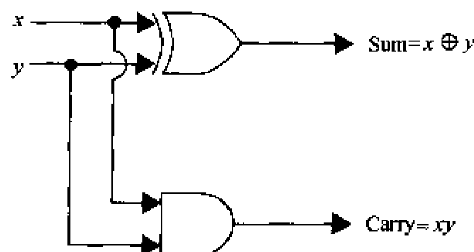
13.

x	y	$x \odot y$	$x \oplus y$	$\overline{(x \oplus y)}$
1	1	1	0	1
1	0	0	1	0
0	1	0	1	0
0	0	1	0	1

15. 是，如真值表所示。

17. a) 6　　b) 5　　c) 5　　d) 6

19.



21. $x_3+x_2\bar{x}_1$

23. 设它具有权 a 和 b，则存在实数 T 使得对 (1, 0) 和 (0, 1)，有 $xa+yb \geqslant T$；但对 (0, 0) 和 (1, 1)，有 $xa+yb<T$。从而 $a \geqslant T$，$b \geqslant T$，$0<T$，且 $a+b<T$。这样 a 和 b 都是正的，这蕴涵 $a+b>a \geqslant T$，矛盾。

第 10 章

10.1 节

1. a) 句子⇒名词短语 不及物动词短语⇒冠词 形容词 名词 不及物动词短语⇒冠词 形容词 名词 不及物动词⇒... (3 步后) ...⇒*the happy hare runs*。

 b) 句子⇒名词短语 不及物动词短语⇒冠词 形容词 名词 不及物动词短语⇒冠词 形容词

名词 不及物动词 副词 ...（5 步后）...⇒*the sleepy tortoise runs quickly*。

c) 句子⇒名词短语 及物动词短语 名词短语⇒冠词 名词 及物动词短语 名词短语⇒冠词 名词 及物动词 名词短语⇒冠词 名词 及物动词 冠词 名词⇒...（5 步后）...⇒*the tortoise passes the hare*。

d) 句子⇒名词短语 及物动词短语 名词短语⇒冠词 形容词 名词 及物动词短语 名词短语⇒冠词 形容词 名词 及物动词 名词短语⇒冠词 形容词 名词 及物动词 冠词 形容词 名词⇒...（6 步后）...⇒*the sleepy hare passes the happy tortoise*。

3. 使得末尾是名词（如 tortoise）只有一个方法，就是将一个名词短语放在末尾，这只能用如下产生式实现：句子→名词短语 及物动词短语 名词短语。而及物动词短语→及物动词→*passes*，但这个句子不含 *passes*。

5. $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111$。

7. a) $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 00S1 \Rightarrow 00S11 \Rightarrow 00S111 \Rightarrow 00S1111 \Rightarrow 001111$。

b) $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 001A \Rightarrow 0011A \Rightarrow 00111A \Rightarrow 001111$。

9. $S \Rightarrow 0SAB \Rightarrow 00SABAB \Rightarrow 00ABAB \Rightarrow 00AABB \Rightarrow 001ABB \Rightarrow 0011BB \Rightarrow 00112B \Rightarrow 001122$。

11. a) $S \to 00S$，$S \to \lambda$。

b) $S \to 10A$，$A \to 00A$，$A \to \lambda$。

c) $S \to AAS$，$S \to BBS$，$AB \to BA$，$BA \to AB$，$S \to \lambda$，$A \to 0$，$B \to 1$。

d) $S \to 0000000000A$，$A \to 0A$，$A \to \lambda$。

e) $S \to AS$，$S \to ABS$，$S \to A$，$AB \to BA$，$BA \to AB$，$A \to 0$，$B \to 1$。

f) $S \to ABS$，$S \to \lambda$，$AB \to BA$，$BA \to AB$，$A \to 0$，$B \to 1$。

g) $S \to ABS$，$S \to T$，$S \to U$，$T \to AT$，$T \to A$，$U \to BU$，$U \to B$，$AB \to BA$，$BA \to AB$，$A \to 0$，$B \to 1$。

13. a) 2 型，非 3 型　b) 3 型，非 2 型　c) 0 型，非 1 型

d) 2 型，非 3 型　e) 2 型　f) 0 型，非 1 型

g) 3 型　h) 0 型，非 1 型　i) 2 型，非 3 型

j) 2 型，非 3 型

15. 设 S_1 和 S_2 分别为 G_1 和 G_2 的初始符，S 是一个新初始符。

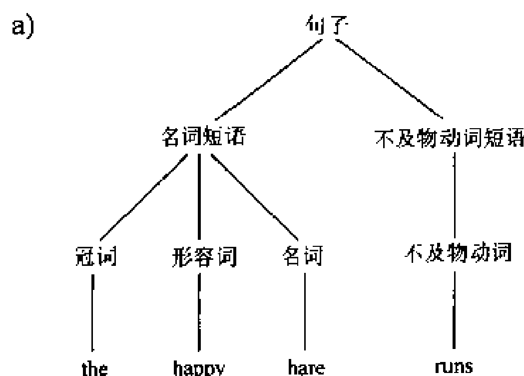
a) 增加 S 和产生式 $S \to S_1$ 及 $S \to S_2$。

b) 增加 S 和产生式 $S \to S_1S_2$。

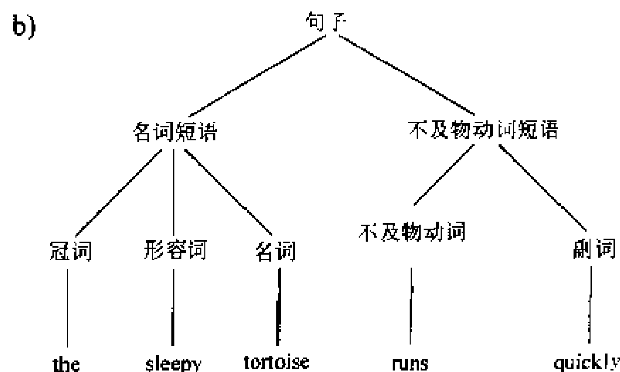
c) 增加 S 和产生式 $S \to \lambda$ 及 $S \to S_1S$。

17.

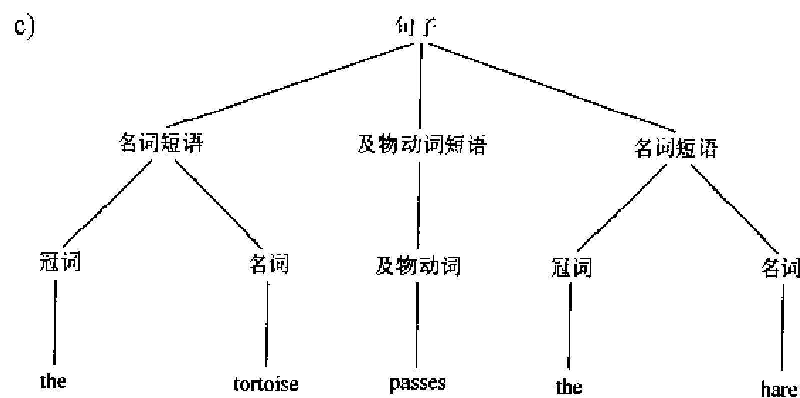
a)



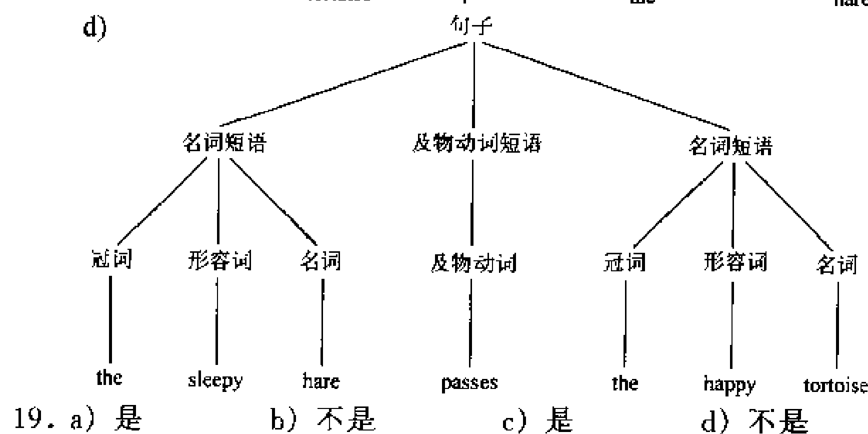
b)



c)

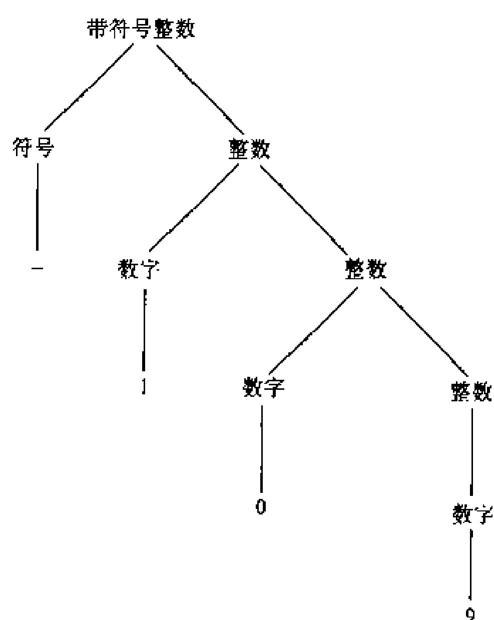


d)



19. a) 是　　b) 不是　　c) 是　　d) 不是

21.



23.

a) S→〈符号〉〈整数〉

S→〈符号〉〈整数〉.〈正整数〉

〈符号〉→ +

〈符号〉→ –

〈整数〉→〈数字〉

〈整数〉→〈整数〉〈数字〉

〈数字〉→ i, i = 1,2,3,4,5,6,7,8,9,0

〈正整数〉→〈整数〉〈非 0 数字〉

〈正整数〉→〈非 0 数字〉〈整数〉

〈正整数〉→〈整数〉〈非 0 数字〉〈整数〉

〈正整数〉→〈非 0 数字〉

〈非 0 数字〉→ i, i = 1,2,3,4,5,6,7,8,9

b) 〈带符号十进制数〉::=〈符号〉〈整数〉|〈符号〉〈整数〉.〈正整数〉

〈符号〉::= + | −

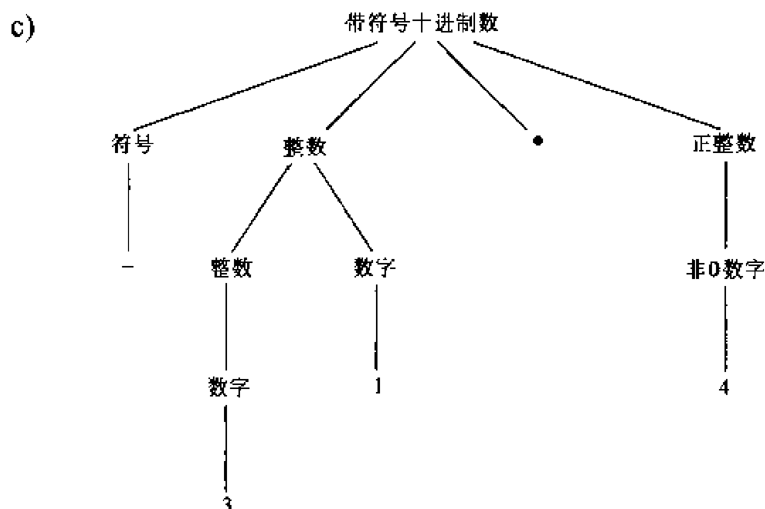
〈整数〉::=〈数字〉|〈整数〉〈数字〉

〈数字〉::= 0|1|2|3|4|5|6|7|8|9

〈非 0 数字〉::= 1|2|3|4|5|6|7|8|9

〈正整数〉::=〈整数〉〈非 0 数字〉|〈非 0 数字〉〈整数〉|〈整数〉〈非 0 数字〉〈整数〉|〈非 0 数字〉

c)

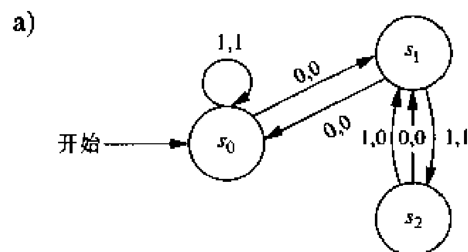


25. $\{(u, v) \mid v$ 可整除 $u\}$

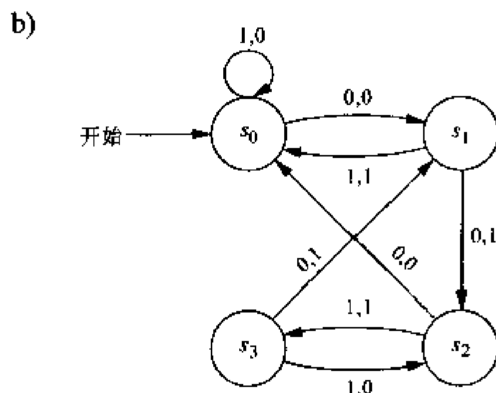
10.2 节

1.

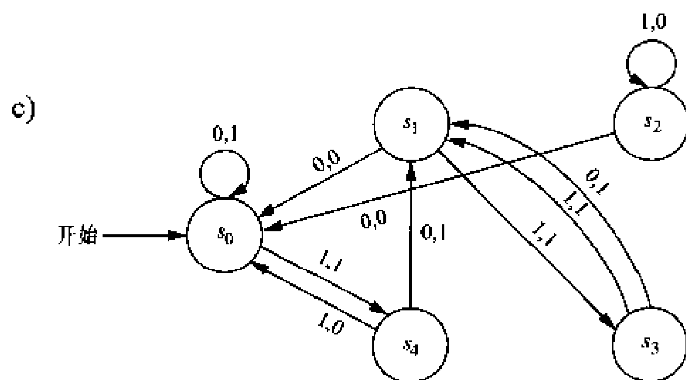
a)



b)

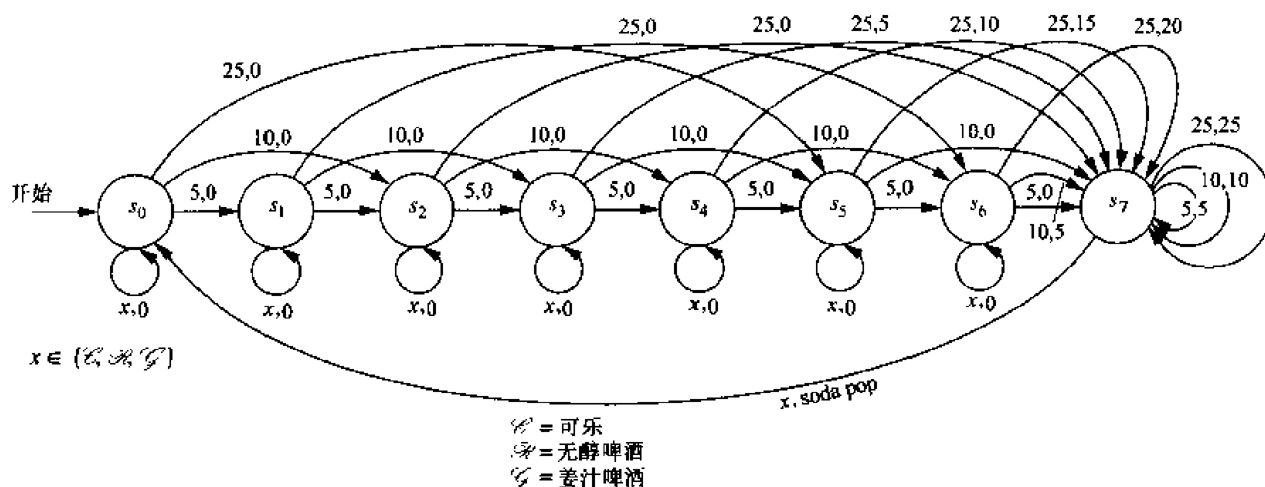


c)

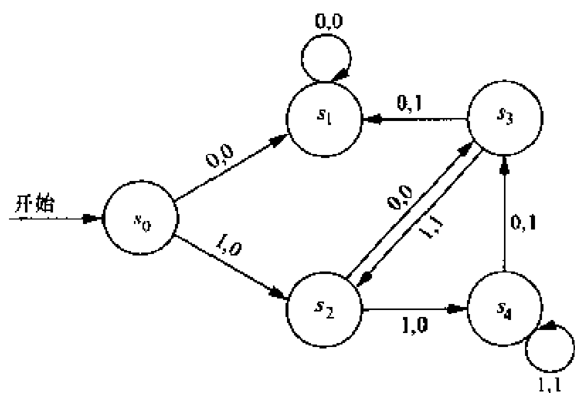


3. a) 1100　　b) 00110110　　c) 11111111111

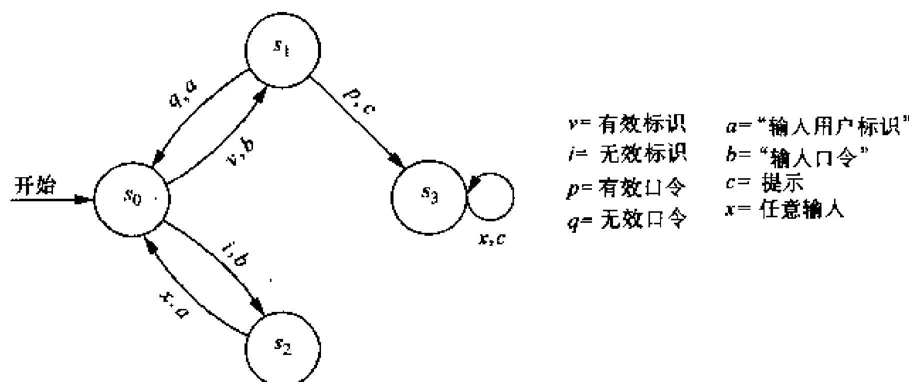
5.



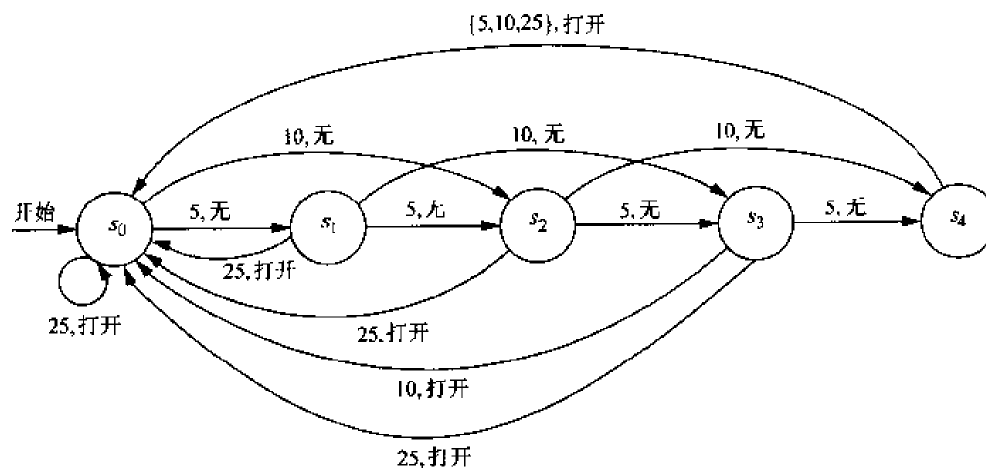
7.



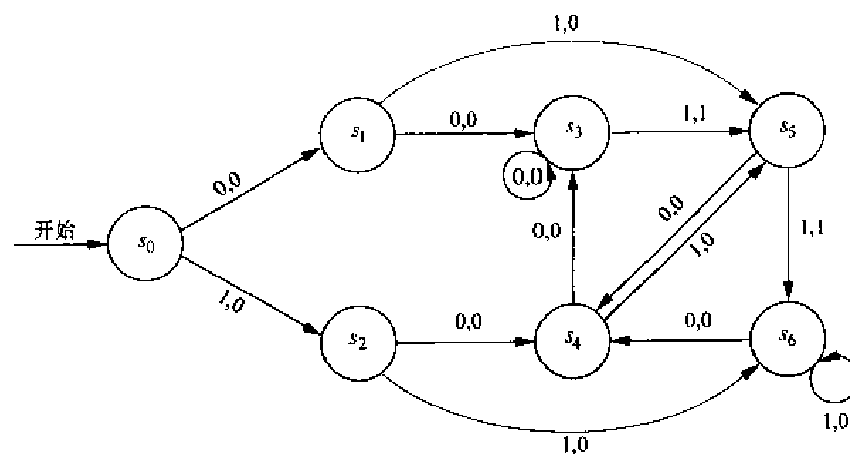
9.



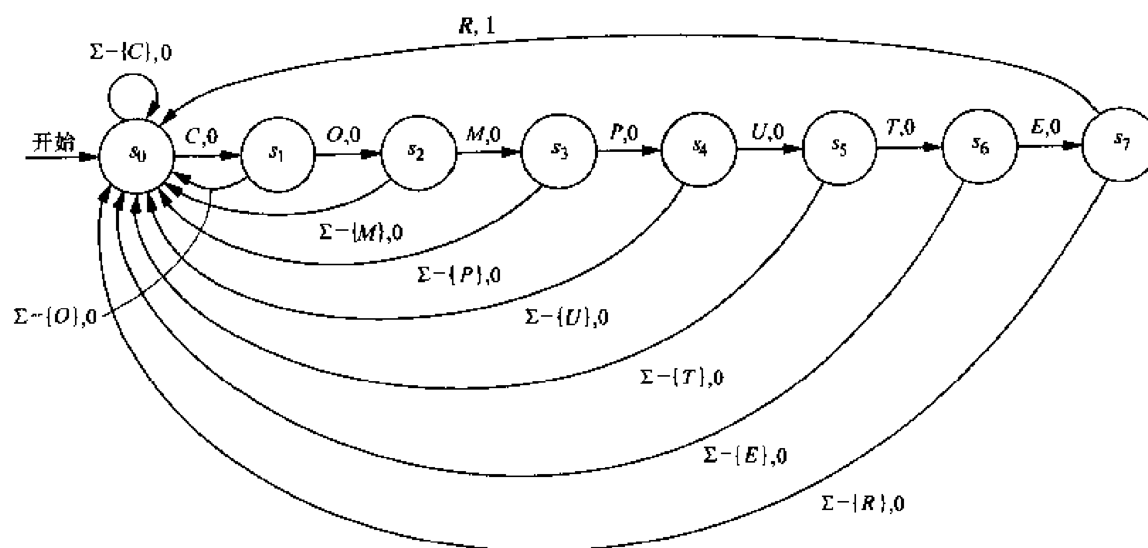
11.


{5,10,25}, 打开
10, 无
10, 无
10, 无
开始
5, 无
5, 无
5, 无
5, 无
25, 打开
25, 打开
25, 打开
10, 打开
25, 打开

13.


1,0
0,0
1,1
0,0
开始
0,0
0,0
1,1
1,0
0,0
1,0
0,0
0,0
1,0
1,0

15.

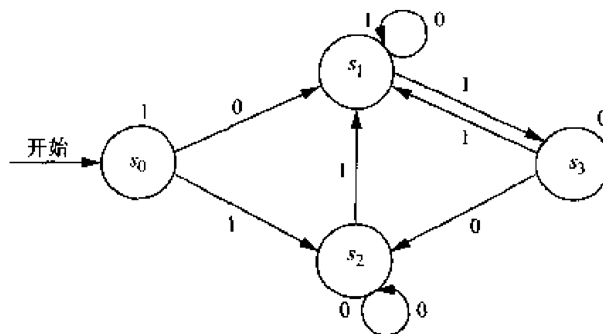

R, 1
Σ−{C},0
开始
C,0
O,0
M,0
P,0
U,0
T,0
E,0
Σ−{M},0
Σ−{P},0
Σ−{O},0
Σ−{U},0
Σ−{T},0
Σ−{E},0
Σ−{R},0

17.

状态	f 输入 0	f 输入 1	g
s_0	s_1	s_2	1
s_1	s_1	s_0	1
s_2	s_1	s_2	0

19. a) 11111　　b) 1000000　　c) 100011001100

21.

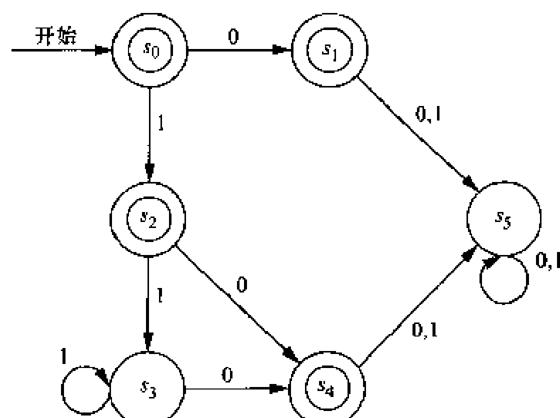


10.3 节

1. a) {000, 001, 1100, 1101}
   b) {000, 0011, 010, 0111}
   c) {00, 011, 110, 1111}
   d) {000000, 000001, 000100, 000101, 010000, 010001, 010100, 010101}
3. $A=\{1, 101\}, B=\{0, 11, 000\}$; $A=\{10, 111, 1010, 1000, 10111, 101000\}, B=\{\lambda\}$; $A=\{\lambda, 10\}, B=\{10, 111, 1000\}$, 或者 $A=\{\lambda\}, B=\{10, 111, 1010, 1000, 10111, 101000\}$。
5. a) 由零个或多个连续的二进制数对 10 构成的串。
   b) 下列由 1 组成的串集合；串中 1 的个数能被 3 整除，包括空串。
   c) 下列串的集合：以 0 开始，且每对 1 之间至少有一个 0。
   d) 下列串的集合：以 1 开始和结尾，且每对 0 之间至少有两个 1。
7. 一个串在 A^* 中当且仅当它是 A 中任意多个串的连接。因为 A 的每个串都在 B 中，故 A^* 的每个串也是 B 中串的连接。从而 $A^* \subseteq B^*$。
9. a) 是　　b) 是　　c) 是
   d) 不是　　e) 是　　f) 是
11. a) 不是　　b) 是　　c) 不是
    d) 不是　　e) 不是　　f) 不是
13. $\{0, 10, 11\}\{0, 1\}^*$
15. $\{0^m 1^n \mid m \geq 0$ 且 $n \geq 1\}$
17. $\{0, 01, 11\}$
19. $\{\lambda, 0\} \cup \{0^m 1^n \mid m \geq 1, n \geq 1\}$

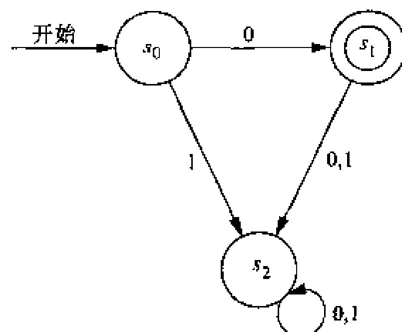
21. $\{10^n \mid n \geq 0\} \cup \{10^n 10^m \mid n, m \geq 0\}$

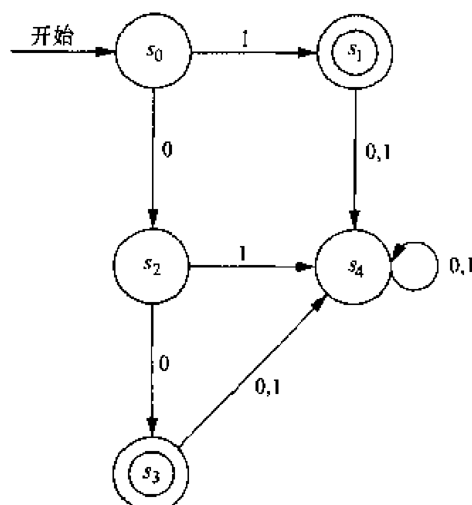
23.

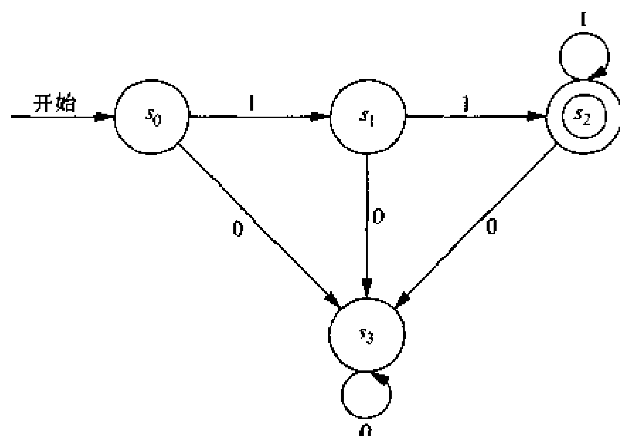


25. 增加非终结状态 s_3 和下列到 s_3 的转移：从 s_0 出发且在输入 0 上、从 s_1 出发且在输入 1 上、从 s_3 出发且在输入 0 或 1 上。

27.







29. 设 M 是一个有限状态自动机，其接受的集合是含有相同 0 和 1 个数的二进制数串组成的集合。且设 M 有 n 个状态。考虑串 $0^{n+1}1^{n+1}$。由鸽巢原理知：当 M 处理这个串时，它在读前面 $n+1$ 个 0 的过程中必定不止一次遇到某个同样的状态。设 s 是一个它至少

遇到两次的状态，则存在一个正整数 k 使得：输入中的 k 个 0 将 M 从状态 s 带回到自身。故 M 在读 $0^{n+1+k}1^{n+1}$ 之后的结束状态与读 $0^{n+1}1^{n+1}$ 之后的结束状态完全相同。因此，由 M 接受 $0^{n+1}1^{n+1}$ 知它也接受 $0^{n+1+k}1^{n+1}$，矛盾。

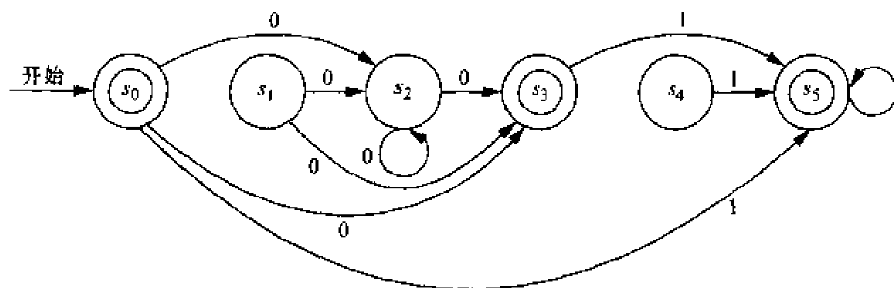
10.4 节

1. a) 任意多个 1，后面跟个 0。
   b) 任意多个 1，后面跟一个或一个以上的 0。
   c) 111 或 011
   d) 任意多个 1 组成的串，或任意多 00 组成的串，或者一列这样的串。
   e) λ，或者是以 1 结尾且在每个 1 之前至少有一个或多个 0 构成的串。
   f) 长度至少为 3 且以 00 结束的串。

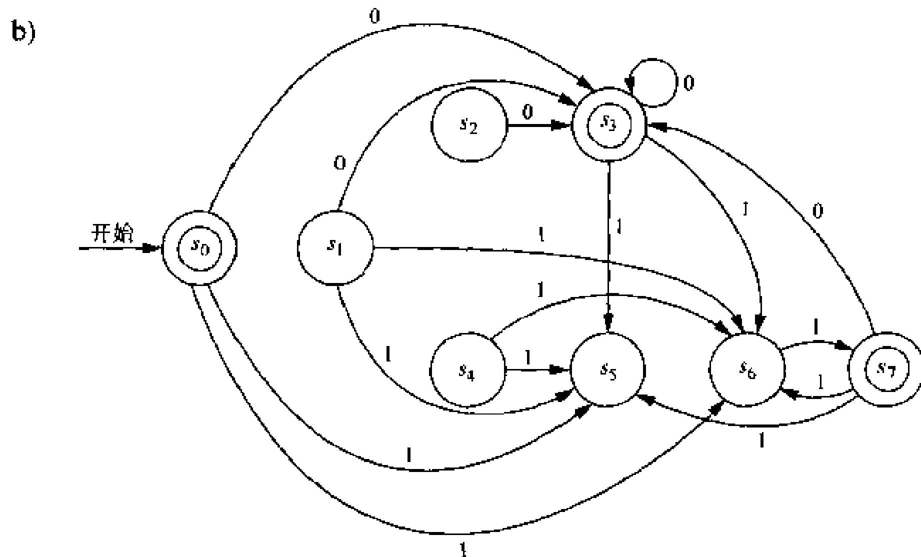
3. a) $\mathbf{00^*1}$　　b) $\mathbf{(0\cup1)(0\cup1)(0\cup1)^*0000^*}$
   c) $\mathbf{0^*1^*\cup1^*0^*}$　　d) $\mathbf{11(111)^*(00)^*}$

5. 用归纳法证明。若 A 的正则表达式是 $\varnothing$、λ 或 x，结论显然成立。否则，设 A 的正则表达式是 **BC**，则 $A=BC$，其中 B 是 **B** 生成的集合，C 是 **C** 生成的集合。由归纳假设，存在正则表达式 **B′**和 **C′**分别生成集合 B^R 和 C^R。因为 $A^R=(BC)^R=C^RB^R$，故 **C′B′**是 A^R 的正则表达式。若 A 的正则表达式是 $\mathbf{B\cup C}$，则 $\mathbf{B'\cup C'}$是 A^R 的正则表达式，因为 $(B\cup C)^R=(B^R)\cup(C^R)$。最后，若 A 的正则表达式是 **B** *，易见：A^R 的正则表达式是(**B′**) * 。

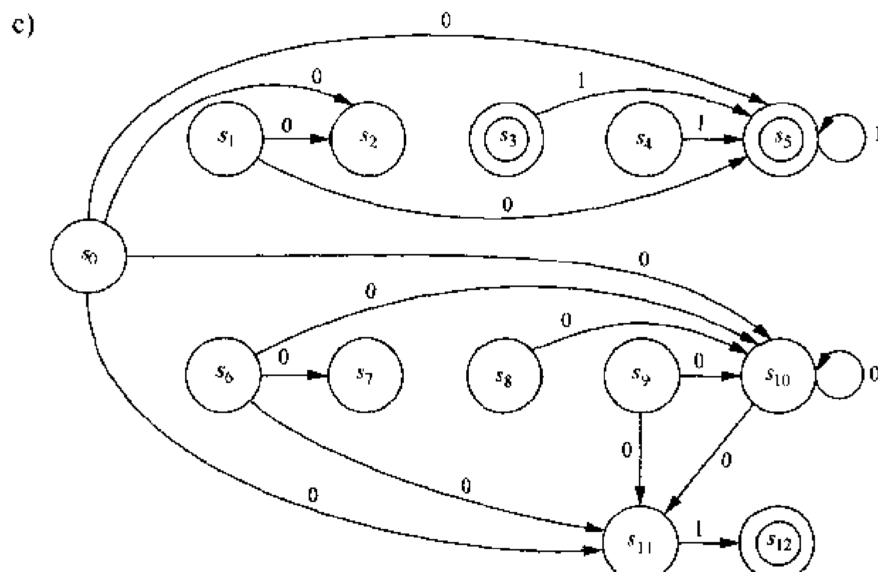
7. a)



b)



c)



9. $S \to 0A$, $S \to 1B$, $S \to 0$, $A \to 0B$, $A \to 1B$, $B \to 0B$, $B \to 1B$。

11. $S \to 0C$, $S \to 1A$, $S \to 1$, $A \to 1A$, $A \to 0C$, $A \to 1$, $B \to 0B$, $B \to 1B$, $B \to 0$, $B \to 1$, $C \to 0C$, $C \to 1B$, $C \to 1$。

13. 这是因为：此自动机中导致终结状态的输入对应于文法中唯一的产生式。

15. 因为 I 是有限的，故必要性显然成立。对于充分性，设状态为 s_{i_0}，s_{i_1}，s_{i_2}，…，s_{i_n}，其中 $n = l(x)$。因为 $n \geqslant |S|$，由鸽巢原理知，某个状态必定要重复。设 y 是 x 中引起循环的部分，即 $x = uyv$，其中 y 将某个 s_j 送到 s_j。则对任意 k，$uy^kv \in L(M)$。从而 $L(M)$是无限的。

17. 设 $L = \{0^{2n}1^n \mid n \geqslant 0\}$ 是正则的，则有一个有限状态机器识别它，令 S 为此机器的状态集。设 $z = 0^{2n}1^n$，其中 $3n \geqslant |S|$。则由泵引理，$z = 0^{2n}1^n = uvw$，$l(v) \geqslant 1$，且 $uv^iw \in \{0^{2n}1^n \mid n \geqslant 0\}$。显然，$v$ 不能既包含 0 又包含 1，因为这样使得 v^2 包含 10。因此 v 全部都是 0 或全部都是 1，这样 uv^2w 包含了太多的 0 或太多的 1，因而它不在 L 中。此矛盾表明 L 不是正则的。

19. 假设 $\{0, 1\}$ 上的回文集是正则的，则有一个有限状态机器识别它，令 S 为此机器的状态集。设 $z = 0^n10^n$，其中 $n > |S|$。由泵引理可找到 u、v 和 w 使 $uv^iw \in L$ 对所有非负整数 i 成立，其中，$l(v) \geqslant 1$，$l(uv) \leqslant |S|$，且 $z = 0^n10^n = uvw$。从而 v 必定是由 0 构成的串（因为 $|n| > |S|$），故 uv^2w 不是回文。因此回文集不是正则的。

10.5 节

1. a) 当机器停机时，带的非空白区域包含串 1111。
   b) 当机器停机时，带的非空白区域包含串 011。
   c) 当机器停机时，带的非空白区域包含串 00001。
   d) 当机器停机时，带的非空白区域包含串 00。

3. 如果带上至少有一个 1，则机器从第一个 1 开始，每隔一个 1 变为 0，且在到达第一个空白符时停机。若一开始的时候带就是空白，则机器不对带作任何改变就停机。如果带的非空白区域所含的都是 0，则机器在相继穿过这些 0 后停机。

5. $(s_0,0,s_1,1,R),(s_0,1,s_0,1,R)$。

7. $(s_0,0,s_0,0,R),(s_0,1,s_1,1,R),(s_1,0,s_1,0,R),(s_1,1,s_1,0,R)$。

9. $(s_0,0,s_1,0,R),(s_0,1,s_0,0,R),(s_1,0,s_1,0,R),(s_1,1,s_0,0,R),(s_1,B,s_2,B,R)$。

11. $(s_0,0,s_0,0,R),(s_0,1,s_1,1,R),(s_1,0,s_1,0,R),(s_1,1,s_0,1,R),(s_0,B,s_2,B,R)$。

13. 若输入串为空串或以 1 开头，则机器在非终结状态 s_0 下停机。若不是，则开头的 0 变为 M，且机器掠过中间所有的 0 和 1，直到到达输入串的尾部或遇到 M，此处它往后退一个方格，并进入状态 s_2。因为可接受串对于左边的每个 0 都必须在右边有个 1，故若此串是可接受的，此处必定有个 1。当这个方格所含的是 1 时，从 s_2 出发的唯一转移就要发生。如果这样，机器就用 M 取代之，且向左继续回走；如果不是只样，则机器在非终结状态 s_2 下停机。在往回走的过程中，如果它所见的是 1，则处于 s_3；如果它所见的是 0，则处于 s_4。最后它处于下列两种情形之一：一是在 s_4 下遇到 1，此时它停机但不接受；或者到达已经在此串初始部分的 0 上写下的最右的 M。若此发生时它处于 s_3，则此串中不再有其他的 0，因此最好也没有其他的 1。这可以用转移(s_3,M,s_5,M,R)和(s_5,M,s_6,M,R)来完成，其中 s_6 是一个终结状态。否则机器在非终结状态 s_5 下停机。若遇到这个 M 时处于 s_4，则事情又完全从头开始，除了将串中剩下最左边的 0 和最右边的 1 用 M 来替代。所以机器在状态 s_4 下移动到剩下最左边的 0，且回到状态 s_0 再重复这个过程。

15. $(s_0,B,s_9,B,L),(s_0,0,s_1,0,L),(s_1,B,s_2,E,R),(s_2,M,s_2,M,R),(s_2,0,s_3,M,R),(s_3,0,s_3,0,R),(s_3,M,s_3,M,R),(s_3,1,s_4,M,R),(s_4,1,s_4,1,R),(s_4,M,s_4,M,R),(s_4,2,s_5,M,R),(s_5,2,s_5,2,R),(s_5,B,s_6,B,L),(s_6,M,s_8,M,L),(s_6,2,s_7,2,L),(s_7,0,s_7,0,L),(s_7,1,s_7,1,L),(s_7,2,s_7,2,L),(s_7,M,s_7,M,L),(s_7,E,s_2,E,R),(s_8,M,s_8,M,L),(s_8,E,s_9,E,L)$，其中 M 和 E 是记号，E 标记输入的左端点。

17. $(s_0,1,s_1,B,R),(s_1,1,s_2,B,R),(s_2,1,s_3,B,R),(s_3,1,s_4,1,R),(s_1,B,s_4,1,R),(s_2,B,s_4,1,R),(s_3,B,s_4,1,R)$。

19. $(s_0,1,s_1,B,R),(s_1,1,s_2,B,R),(s_1,B,s_6,B,R),(s_2,1,s_3,B,R),(s_2,B,s_6,B,R),(s_3,1,s_4,B,R),(s_3,B,s_6,B,R),(s_4,1,s_5,B,R),(s_4,B,s_6,B,R),(s_6,B,s_{10},1,R),(s_5,1,s_5,B,R),(s_5,B,s_7,1,R),(s_7,B,s_8,1,R),(s_8,B,s_9,1,R),(s_9,B,s_{10},1,R)$。

21. $(s_0,0,s_0,0,R),(s_0,*,s_5,B,R),(s_3,*,s_3,*,L),(s_3,0,s_3,0,L),(s_3,1,s_3,1,L),(s_3,B,s_0,B,R),(s_5,1,s_5,B,R),(s_5,0,s_5,B,R),(s_5,B,s_6,B,L),(s_6,B,s_6,B,L),(s_6,0,s_7,1,L),(s_7,0,s_7,1,L),(s_0,1,s_1,0,R),(s_1,1,s_1,1,R),(s_1,*,s_2,*,R),(s_2,0,s_2,0,R),(s_2,1,s_3,0,L),(s_2,B,s_4,B,L),(s_4,0,s_4,1,L),(s_4,*,s_8,B,L),(s_8,0,s_8,B,L),(s_8,1,s_8,B,L)$。

23. $(s_0,B,s_1,1,L),(s_0,1,s_1,1,R),(s_1,B,s_0,1,R)$。

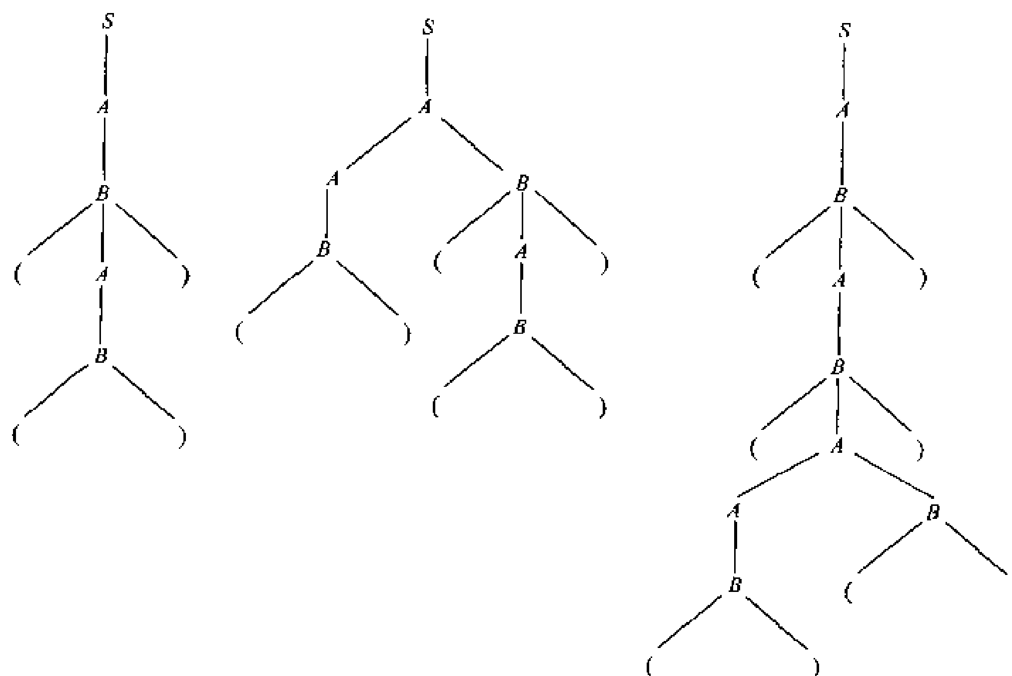
补充练习

1. a) $S\to 00S111$，$S\to\lambda$。

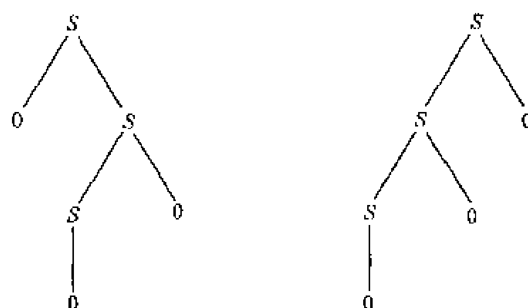
   b) $S\to AABS$，$AB\to BA$，$BA\to AB$，$A\to 0$，$B\to 1$，$S\to\lambda$。

   c) $S\to ET$，$T\to 0TA$，$T\to 1TB$，$T\to\lambda$，$0A\to A0$，$1A\to A1$，$0B\to B0$，$1B\to B1$，$EA\to E0$，$EB\to E1$，$E\to\lambda$。

3.



5.

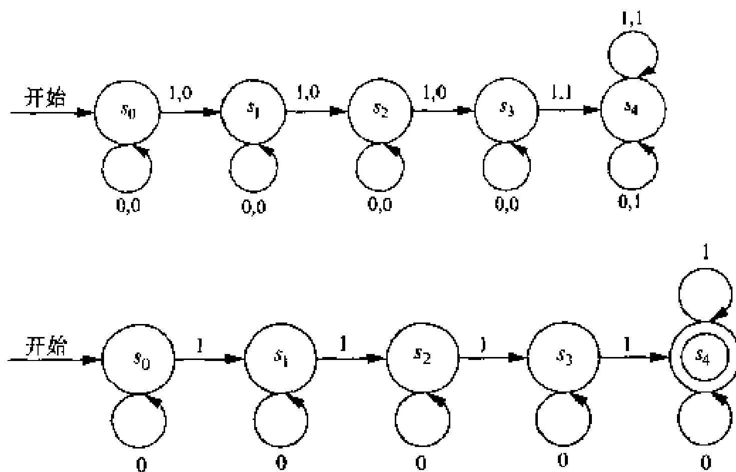


7. 不是。取 $A=\{1,10\}$，$B=\{0,00\}$。

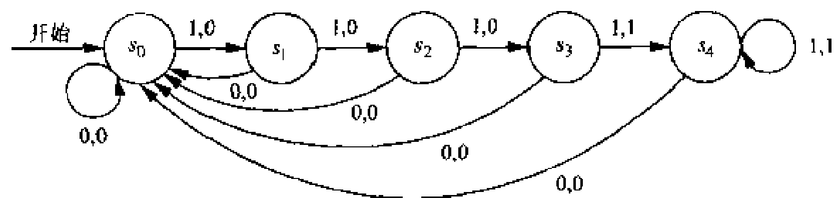
9. 不是。取 $A=\{00,000,00000\}$，$B=\{00,000\}$。

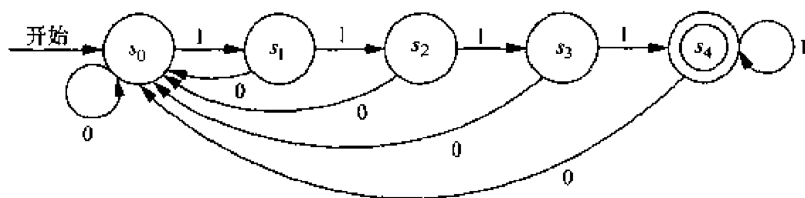
11. a) 1　　b) 1　　c) 2
    d) 3　　e) 2　　f) 4

13.



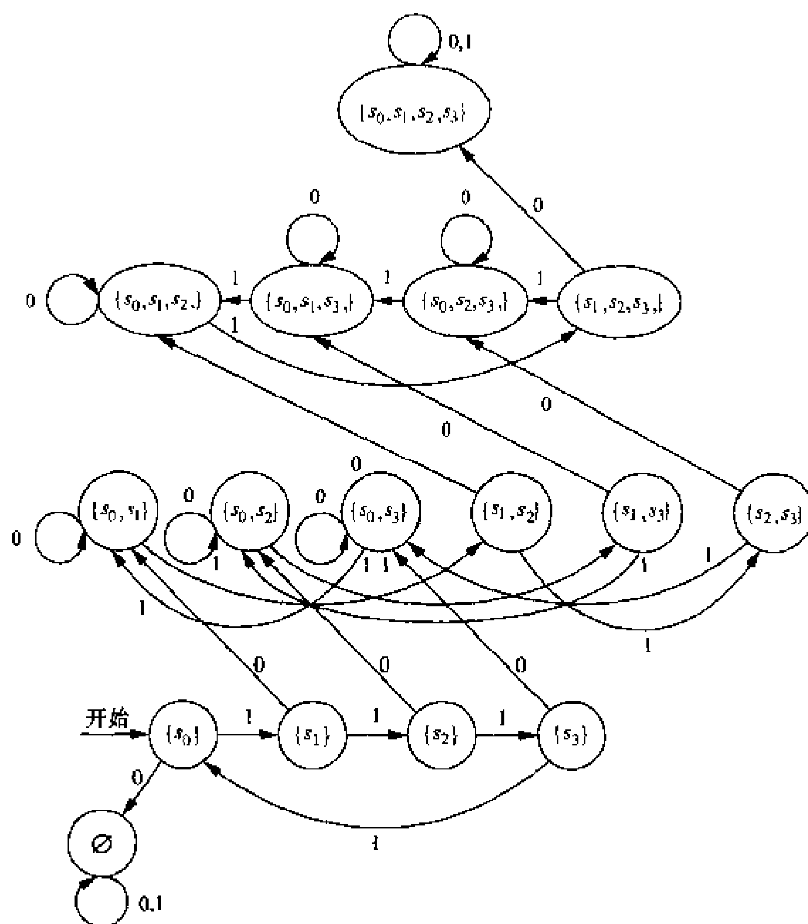
15.



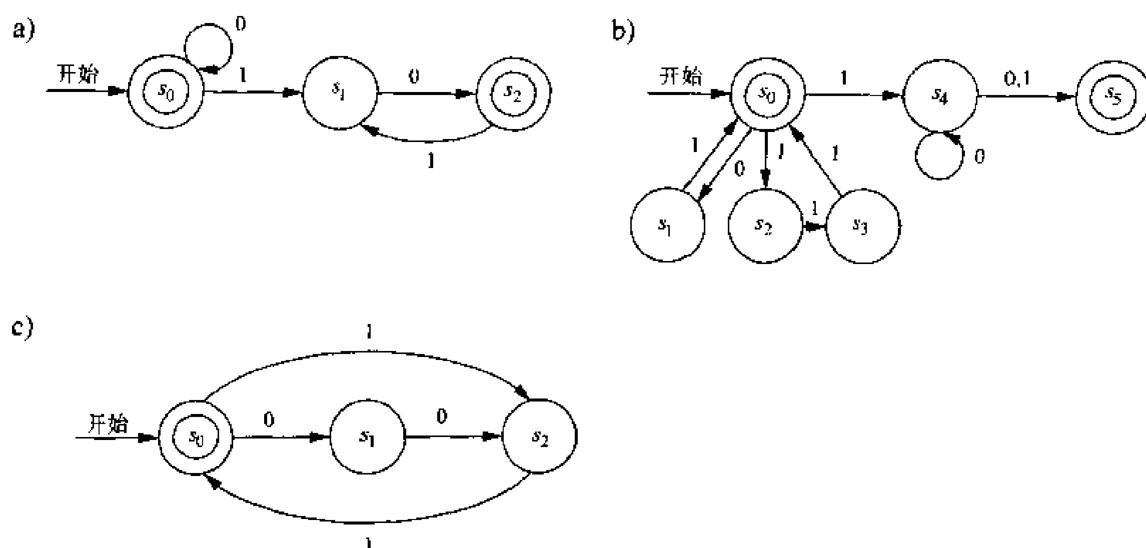


17. a) $n^{nk+1}m^{nk}$　　　b) $n^{nk+1}m^{n}$

19.

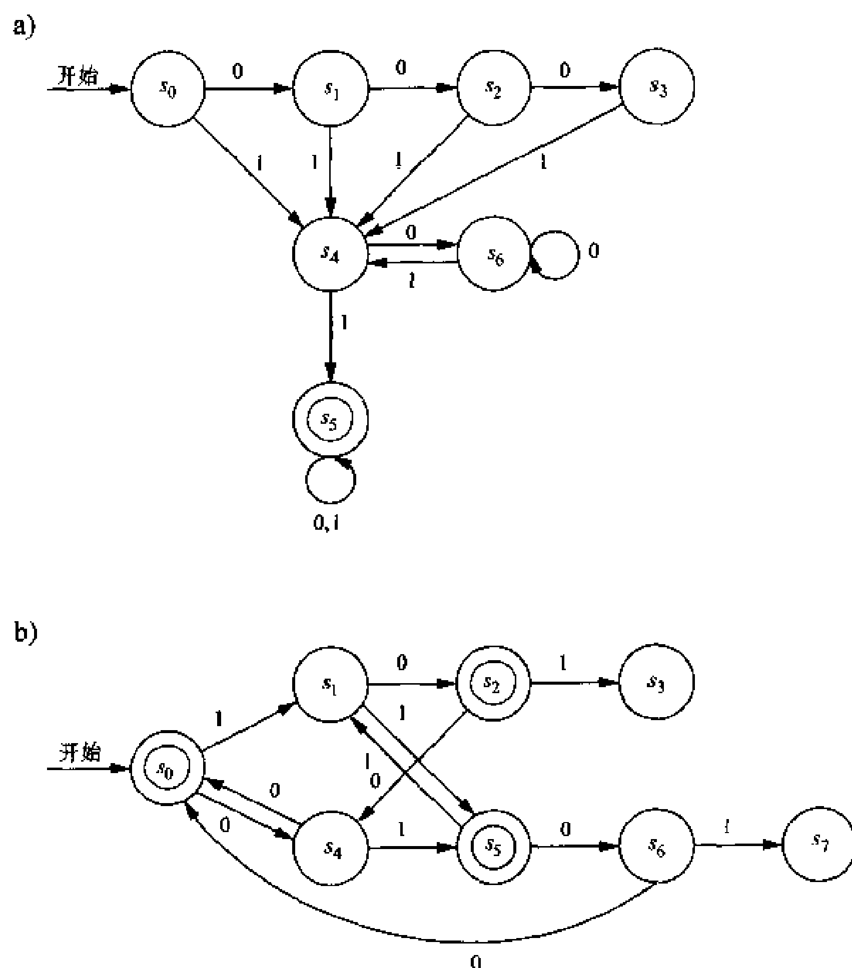


21.

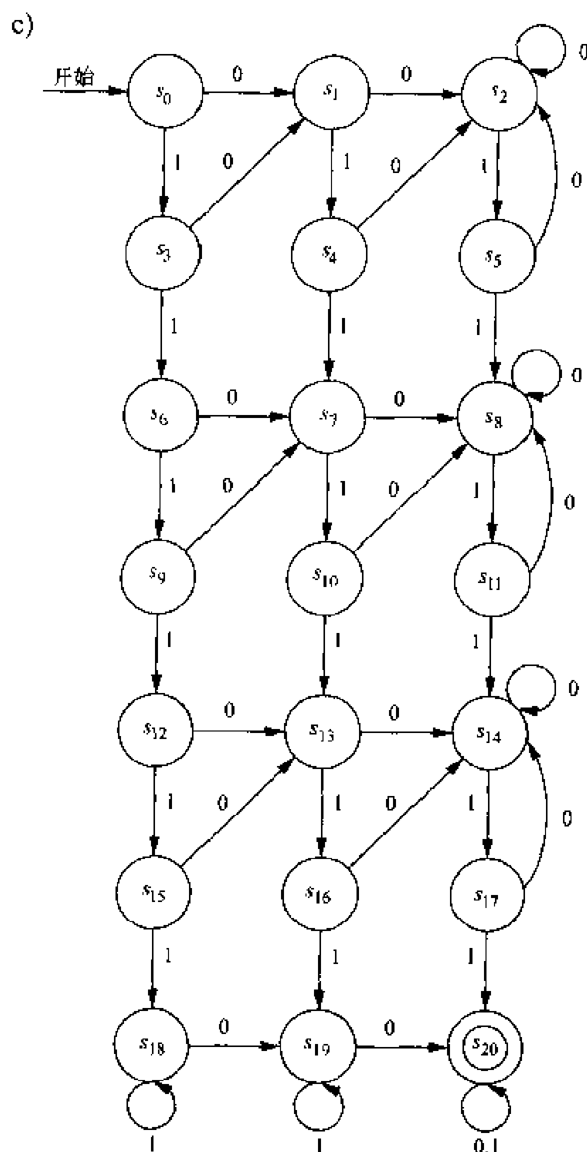


23. 用状态 S 和终结状态 F 构造 A 的确定型有限状态自动机。对 $\overline{A}$ 使用相同的自动机，但其终结状态为 $S-F$。

25.



c)



27. 假设$L=\{1^p|p$ 是素数$\}$是正则的，则有一个有限状态机器识别它．令 S 为此机器的状态集。设 $z=1^p$，其中 p 是一个大于$|S|$的素数（这样的素数是存在的，因为有无限多个素数)。由泵引理知：必定能够将 z 写成 $z=uvw$，且 $l(uv)\leqslant|S|$，$l(v)\geqslant 1$。并且对任意非负整数 i，$uv^iw\in L$。因为 z 是由 1 构成的串，可令 $u=1^a$，$v=1^b$，$w=1^c$，其中 $a+b+c=p$，$a+b\leqslant n$，且 $b\geqslant 1$。这意味着 $uv^iw=1^a1^{bi}1^c=1^{(a+b+c)+b(i-1)}=1^{p+b(i-1)}$。现在取 $i=p+1$，则 $uv^iw=1^{p(1+b)}$。因为$p(1+b)$不是素数，故 $uv^iw\notin L$，矛盾。

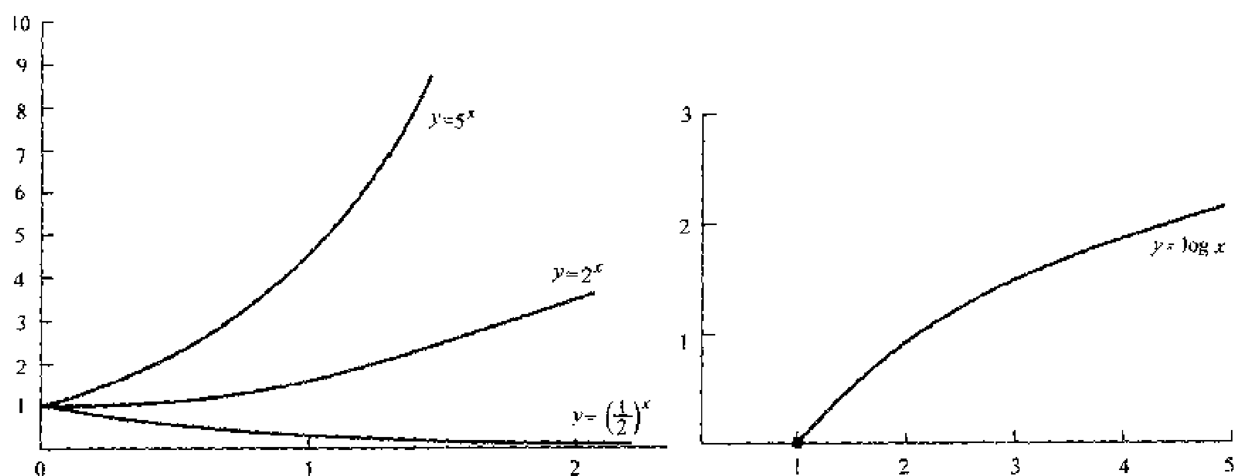
附录

附录 A

1. a) 2^3　　b) 2^6　　c) 2^4

3. a) $2y$　　b) $2y/3$　　c) $y/2$

5.



附录 B

1. 第一块执行后，b 原来的值被赋给 a，c 原来的值被赋给 b。但第二块执行之后，c 原来的值被赋给 b，且 c 原来的值也被赋给了 a。

3. 下面的 **while** 结构实现相同的事情：

```
i := 起始值
while i ≤ 结束值
begin
    语句
    i := i + 1
end
```

推荐读物

提出的书面材料和相关的网站都是用来进一步学习本书内容的资源。本推荐读物描述的书面材料，将分章列出，并与特定研究课题相联系。有一些内容全面的参考书特别值得一提。读者可能发现，Rosen 的《离散和组合数学手册》(*Handbook of Discrete and Combinatorial Mathematics*)[Ro99]非常有用，这是一本综合参考书。在 Michaels 和 Rosen[MiRo9]中，可以找到离散数学的其他应用。在 Gruska [Gr97] 中，有计算机科学中许多课题（包括本书中讨论的课题）的更深入介绍。本书提到的许多有关数学和计算机科学的文献信息可以在 Gillispie[Gi70]中查到。

要查找相关的网站，请在本书的网站中查询“离散数学 Web 指南”。此网站的统一资源定位器（URL）是www. mhhe. com/rosen。

第 1 章

学习逻辑的一个有趣方法是读Lewis Carroll[Ca78]一书。逻辑的一般参考书包括Mendelson[Me64]、Stoll[St74]和 Suppes[Su87]。Gries 和Schneider[GrSc93]中可以找到离散数学中的逻辑学的全面介绍。Lin 和 Lin [LiLi8]是关于集合论及其应用的一本易读的教材。公理集合论的进展可在Halmos[Ha60]、Monk[Mo60]或 Stoll[St74]中找到。Brualdi[Br77]或 Reingold、Nievergelt 和 Deo [ReNiDe77]中有关于多重集合的介绍。Negoita[Ne85]和 Zimmerman [Zi91]都论述了模糊集合及其在专家系统和人工智能中的应用。微积分的书中，如 Apostol [Ap67]、Spivak[Sp94]及 Thomas 与 Finney[ThFi96]，都有关于函数的讨论。有关整数序列的最好书面材料是 Sloane 和Plouffe[FlPl95]。Stanat 和McAllister[StMc77]中有一整节讨论可数性。在 Knuth[Kn97a]中可以找到函数大 O 估计的广泛讨论。在 Arbib、Kfoury 与Moll [ArKfMo80]，Bobrow 与Arbib[BoAr74]，Beckman[Be80]和 Tremblay 与 Manohar[TrMa75]中，都可找到有关计算机科学的数学基础的讨论。本书提及了许多数学家和计算机科学家，他们的传记可在 Gillispie [Gi70] 找到。

第 2 章

Knuth 的文章[Kn77]与 Wirth 的文章[Wi84]是算法这门学科的入门文献。有关算法及其复杂性的一般参考书有 Aho、Hopcroft 和 Ullman [AhHoUl74]，Baase[Ba88]，Cormen、Leierson 和 Rivest [CoLeRi90]，Gonnet[Go84]，Goodman 和 Hedetniemi[GoHe77]，Harel [Ha87]，Horowitz 和 Sahni[HoSa82]，Knuth 关于计算机程序设计艺术的著名系列著作 [Kn97a, b, 98]，Kronsjö [Kr87]，Pohl 和 Shaw [PoSh8]，Purdom 和 Brown [PuBr85]，Sedgewick[Se88]，Wilf[Wi86]以及 Wirth[Wi76]。有关数论的参考书有 Hardy 和 Wright [HaWr79]，Leveque[Le77]，Rosen[Ro93]以及 Stark [St78]。介绍数论在密码学中应用的书有 Denning[De82]，Menezes、van Oorschot 和Vanstone[MeOoVa97]，Rosen[Ro93]，Seberry 和 Pieprzyk[SePi89]，Sinkov[Si66]以及 Stinson[St95]。Rivest、Shamir 和 Adleman

[RiShAd78]介绍了 RSA 公钥系统。Knuth[Kn97b]以及 Pohl 与 Shaw 的[PoSh81]中讨论了有关计算机算术的算法。所有的线性代数书中都有矩阵及其运算的介绍，如Curtis[Cu84]和Strang[St88]。

第 3 章

Pólya 的三本书[Po57、Po62、Po54]以令人愉快的方式讨论了构造证明的科学与艺术。在Sominskii[So61]的英语译本(原文为俄文)中可以找到数学归纳法的通俗介绍。Golomb[Go94]中讨论了用 L 形片或其他类型的片来覆盖一个棋盘中的部分或全部方格问题。包含数学归纳法和递归定义全面介绍的书有 Liu[Li85]，Sahni[Sa85]，Stanat 和 McAllister[StMc77]以及 Tremblay 和 Manohar 的[TrMa75]。阿克曼函数是由阿克曼(Ackermann)在1922 年引入的，它起源于递归函数论(例如可参见 Beckman[Be80]和 McNaughton[Mc82])和某些集合论算法的复杂性分析(见 Tarjan[Ta83])。 Roberts[Ro86]、Rohl[Ro84]和 Wand[Wa80]都讨论了递归。程序正确性的讨论以及用来证明程序是正确的逻辑机制的讨论可在Alagic 和 Arbib[AlAr78]、Anderson[An79]、Backhouse[Ba86]、Sahni[Sa85]以及 Stanat 和McAllister[StMc77]中找到。

第 4 章

计数技术及其应用的一般参考书有 Anderson [An74]、Berman 和 Fryer [BeFr72]、Bogart [Bo86]、Bose 和 Manvel [BoMa86]、Brualdi [Br97]、Cohen [Co78]、Grimaldi [Gr94]、Liu [Li68]、Pólya、Tarjan 和 Woods [PoTaWo83]，Riordan [Ri58]、Roberts [Ro84]、Tucker [Tu85] 以及 Williamson [Wi85]。Vilenkin [Vi71] 中有选择地介绍了一些组合问题及它们的解，在 Lovász[Lo79]中可以找到一些更困难的组合问题。关于因特网协议寻址和数据报的信息可在 Comer [Co95] 中找到。鸽巢原理的应用可在 Brualdi [Br77]、Liu[Li85]和 Roberts[Ro84]中找到。离散概率论的参考书有 Feller [Fe68] 和 Ross [Ro84]。在Riordan[Ri68]中可以找到许多组合数学中的恒等式。Even [Ev73]、Lehmer [Le64] 以及Reingold、Nievergelt 和 Deo [ReNiDe77] 都描述了组合算法（包括生成置换和组合的算法）。

第 5 章

在 Roberts[Ro84]和 Tucker[Tu85]中可以找到使用递推关系的许多不同模型。在 Brualdi[Br77]、Liu[Li85]和 Mattson[Ma93]中可以找到常系数线性齐次递推关系以及相关的非齐次递推关系的详尽介绍。Roberts[Ro84]及 Stanat 和 McAllister[StMc77]都介绍了“分而治之”算法及其复杂性。Aho、Hopcroft 和Ullman[AhHoUl74]及 Knuth[Kn81]描述了更快的整数乘法和矩阵乘法。Pólya、Tarjan 和Woods[PoTaWo83]中有生成函数的极好介绍。详细研究生成函数的文献有Brualdi[Br77]、Cohen[Co78]、Graham、Knuth 和 Patashnik[GrKnPa94]、Grimaldi[Gr94]和 Roberts[Ro84]。容斥原理的其他应用可在 Liu[Li85, Li68]、Roberts[Ro84]和 Ryser[Ry63]中找到。

第 6 章

关于关系（包括等价关系和偏序）的通用参考书有 Bobrow 和Arbib[BoAr74]、Grimaldi

[Gr94]、Sanhi[Sa85]以及 Tremblay 和 Manohar[TrMa75]。Date[Da82]讨论了数据库的关系模型。Roy 和 Warshall 求传递闭包的原始文献分别在[Ro59]和[Wa62]中。研究有向图的文献有 Chartrand 和Lesniak[ChLe86]、Grimaldi[Gr94]、Robinson 和 Foulds[RoFo86]、Roberts[Ro84]以及 Tucker[Tu85]。 Denning[De82]讨论了格在信息流处理中的应用。

第 7 章

图论的一般参考书有 Behzad 和Chartrand[BeCh71]、Chartrand 和 Lesniak[ChLe96]、Bondy 和 Murty[BoMu76]、Chartrand 和 Oellermann[ChOe93]、Graver 和 Watkins[GrWa77]、Grimaldi[Gr94]、Gross 和 Yellen[GrYe99]、Harary[Ha69]、Ore[Or63]、Roberts[Ro84]、Tucker[Tu85]、Wilson[Wi85]以及 Wilson 和 Watkins[WiWa90]。介绍图论广泛应用文献的有Chartrand[Ch77]、Deo[De74]、Foulds[Fo92]、Roberts[Ro84、Ro76]、Wilson 和 Beineke[WiBe79]以及 McHugh[Mc90]。 Gibbons[Gi85]全面描述了图论算法。图论算法的其他参考书还有Buckley 和 Harary[BuHa90]，Chartrand 和 Oellermann[ChOe93]、Chachra、Ghare 和 Moore[ChGhMo79]、Even[Ev73、Ev79]、Hu[Hu82]以及 Reingold、Nievergelt 和 Deo[ReNiDe77]。[Eu53]是欧拉关于哥尼斯堡桥问题原文的译文。 Gibbons[Gi85]、Liu[Li85]以及 Reingold、Nievergelt 和 Deo[ReNiDe77]都讨论了 Dijkstra 算法、Dijkstra 的原文在[Di59]中。Kuratwoski 定理的证明可在Harary[Ha69]和 Liu[Li68]中找到。Chartrand 和Lesniak[ChLe96]研究了图的交叉数和厚度。有关图着色及四色定理的参考书有Barnette[Ba83]及 Saaty 和 Kainen[SaKa86]。Appel 和 Haken[ApHa76]报道了四色定理的最初结果。Roberts[Ro84]描述了图着色的应用。Biggs、Lloyd 和 Wilson[BiLlWi86]介绍图论的历史。Akl[Ak89]与 Siegel 和 Hsu[SiHs88]讨论了并行处理的互连网络。

第 8 章

下列文献都讨论了树：Deo [De74]、Grimaldi [Gr94]、Knuth [Kn97a]、Roberts [Ro84]、Tucker [Tu85]。Gotlieb 和 Gotlieb [GoGo78]、Horowitz 和 Sahni[HoSa82]以及 Knuth[Kn97a,98]描述了树在计算机科学中的应用。Roberts[Ro84]介绍了树在许多不同领域中的应用。Hamming[Ha80]介绍了前缀码和霍夫曼码。Knuth[Kn98]详细讨论了存储与搜索算法及其复杂性。回溯法是一种旧方法，它在解迷宫难题中的应用可从[Lu91]中找到，这是 Lucas 于 1891 年写的一本书。Reingold、Nievergelt 和Deo[ReNiDe77]广泛讨论了怎么用回溯法来求解问题。Gibbons[Gi85]和 Reingold、Nievergelt 和Deo[ReNiDe77]还讨论了构造生成树和极小生成树算法。Graham 和Hell[GrHe85]介绍了求极小生成树算法的历史和背景。Prim 和 Kruskal 分别在[Pr57]和[Kr56]中描述了他们的求极小生成树算法。Sollin 算法是一个非常适用于并行处理算法的例子。虽然 Sollin 从未发表这个算法的描述，但Even[Ev73]及 Goodman 和Hedetniemi[GoHe77]都描述了它。

第 9 章

讨论布尔代数的文献有 Grimaldi [Gr94]、Hohn [Ho66]、Kohavi[Ko86]以及 Tremblay 和 Manohar [TrMa75]。Hohn[Ho66]和 Kohavi[Ko86]描述了布尔代数在逻辑电路和开关电路中的应用。用图来处理乘积的和展开式的极小化的原始文献是 Karnaugh [Ka53] 和

Veitch [Ve52]。奎因-莫可拉斯基方法是在McCluskey[Mc56]和 Quine[Qu52,Qu55]中引入的。Kohavi[Ko78]介绍了阈值函数。

第 10 章

语言和自动机理论的一般参考书有 Denning、Dennis 和 Qualitz [DeDeQu81]、Hopcroft 和 Ullman [HoUl79]、Hopkin 和 Moss [HoMo76]、Lewis 和 Papadimitriou [LePa84] 以及 McNaughton [Mc82]。米利机和摩尔机是在Mealy[Me55]和 Moore[Mo56]中引入的。克莱因定理的原始证明可在[Kl56]中找到。下列文献讨论了更强大的计算模型（包括下推自动机和图灵机）：Brookshear [Br89]、Hennie [He77]、Hopcroft 和 Ullman [HoUl79]、Hopkin 和 Moss [HoMo76]、Martin [Ma91] 以及 Wood [Wo87]。在Herken[He88]中可以找到一些关于图灵机及相关机器的历史和应用的有趣文章。忙碌海狸机器首先由 Rado 在[Ra62]中引入，有关这些机器的信息可在下列文献中找到：Dewdney [De84, 93]、Herken[He88]中由Brady 所写的文章以及 Wood [Wo87]。

附录

指数和对数函数的详细介绍可在微积分书中找到。如 Apostol [Ap67]、Spivak [Sp80] 及 Thomas 和 Finney [ThFi92]。附录 2 中描述的伪代码与 Pascal 十分相像。Pohl 和Shaw [PoSh81]用类似形式语言描述算法。Writh[Wr76]描述了怎么用算法和数据结构来构造 Pascal 程序。Rohl[Ro84]说明怎么用 Pascal 来实现递归程序。

参 考 文 献

[**AhHoUl74**] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[**Ak89**] S. G. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[**AlAr78**] S. Alagic and M. A. Arbib, *The Design of Well-Structured and Correct Programs*, Springer-Verlag, New York, 1978.

[**An74**] I. Anderson, *A First Course in Combinatorial Mathematics*, Clarendon, Oxford, England, 1974.

[**An79**] R. B. Anderson, *Proving Programs Correct*, Wiley, New York, 1979.

[**Ap67**] T. M. Apostol, *Calculus*, Vol.1, 2d ed., Wiley, New York, 1967.

[**ApHa76**] K. Appel and W. Haken, "Every Planar Map Is 4-colorable," *Bulletin of the AMS*, 82 (1976), 711–712.

[**ArKfMo80**] M. A. Arbib, A. J. Kfoury, and R. N. Moll, *A Basis for Theoretical Computer Science*, Springer-Verlag, New York, 1980.

[**Ba88**] S. Baase, *Computer Algorithms*, 2d ed., Addison-Wesley, Reading. MA, 1988.

[**Ba86**] R. C. Backhouse, *Program Construction and Verification*, Prentice Hall, Englewood Cliffs, NJ, 1986.

[**Ba83**] D. Barnette, *Map Coloring*, *Polyhedra*, *and the Four-Color Problem*, Mathematical Association of America, Washington, D.C., 1983.

[**Be80**] F. S. Beckman, *Mathematical Foundations of Programming*, Addison-Wesley, Reading, MA, 1980.

[**BeCh71**] M. Behzad and G. Chartrand, *Introduction to the Theory of Graphs*, Allyn & Bacon, Boston, 1971.

[**BeFr72**] G. Berman and K. D. Fryer, *Introduction to Combinatorics*, Academic Press, New York, 1972.

[**BiLlWi86**] N. L. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory* 1736–1936, Clarendon, Oxford, England, 1986.

[**BoAr74**] L. S. Bobrow and M. A. Arbib, *Discrete Mathematics*, Saunders, Philadelphia, 1974.

[**Bo86**] K. P. Bogart, *Introductory Combinatorics*, Wiley, New York, 1986.

[**BoMu76**] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North-Holland, New York, 1976.

[**BoMa86**] R. C. Bose and B. Manvel, *Introduction to Combinatorial Theory*, Wiley, New York, 1986.

[**Br89**] J. G. Brookshear, *Theory of Computation*, Benjamin Cummings, Redwood City, CA, 1989.

[**Br97**] R. A. Brualdi, *Introductory Combinatorics*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1997.

[**BuHa90**] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley, Redwood City, CA, 1990.

[**Ca78**] L. Carroll, *Symbolic Logic*, Crown, New York, 1978.

[**ChGhMo79**] V. Chachra, P. M. Ghare, and J. M. Moore, *Applications of Graph Theory Algorithms*, North-Holland, New York, 1979.

[**Ch77**] G. Chartrand, *Graphs as Mathematical Models*, Prindle, Weber & Schmidt, Boston, 1977.

[**ChLe86**] G. Chartrand and L. Lesniak, *Graphs and Digraphs*, 3d ed., Wadsworth, Belmont, CA, 1996.

[**ChOe93**] G. Chartrand and O. R. Oellermann, *Applied Algorithmic Graph Theory*, McGraw-Hill, New York, 1993.

[**Co78**] D. I. A. Cohen, *Basic Techniques of Combinatorial Theory*, Wiley, New York, 1978.

[**Co95**] D. Comer, Internetworking with TCP/IP, *Principles*, *Protocols*, *and Architecture*, vol.1, 3d ed., Prentice Hall, Englewood Cliffs, NJ, 1995.

[**CoLeRi90**] T. H. Cormen, C. E. Leierson, R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[**Cu84**] C. W. Curtis, *Linear Algebra*, Springer-Verlag, New York, 1984.

[**Da82**] C. J. Date, *An Introduction to Database Systems.3d* ed., Addison-Wesley, Reading, MA, 1982.

[**De82**] D. E. R. Denning, *Cryptography and Data Security*, Addison-Wesley, Reading, MA, 1982.

[**DeDeQu81**] P. J. Denning, J. B. Dennis, and J. E. Qualitz, *Machines*, *Languages*, *and Computation*, Prentice Hall, Englewood Cliffs, NJ.1981.

[**De74**] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1974.

[**De84**] A. K. Dewdney, "Computer Recreations," *Scientific American*, 251, *no*.2 (August 1984), 19–23; 252, no.3 (March 1985), 14–23; 251, no.4 (April 1985), 20–30.

[**De93**] A. K. Dewdney, *The New Turing Omnibus*: *Sixty-Six Excursions in Computer Science*, W. H. Freeman, New York, 1993.

[**Di59**] E. Dijkstra, "Two Problems in Connexion with Graphs," *Numerische Mathematik*, 1 (1959), 269–271.

[**Eu53**] L. Euler, "The Koenigsberg Bridges," *Scientific American*, 189, no.1 (July 1953), 66–70.

[**Ev73**] S. Even, *Algorithmic Combinatorics*, Macmillan, New York, 1973.

[**Ev79**] S. Even, *Graph Algorithms*, Computer Science Press, Rockville, MD, 1979.

[**Fe68**] W. Feller, *An Introduction to Probability Theory and Its Applications*, *vol.1*, *3d ed.*, Wiley, New York, 1968.

[**Fo92**] L. R. Foulds, *Graph Theory Applications*, Springer-Verlag, New York, 1992.

[**Gi85**] A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, Cambridge, England, 1985.

[**Gi70**] C. C. Gillispie, ed., *Dictionary of Scientific Biography*, Scribner's, New York, 1970.

[**Go94**] S. W. Golomb, *Polyominoes*, Princeton University Press, Princeton, NJ, 1994.

[**Go84**] G. H. Gonnet, *Handbook of Algorithms and Data Structures*, Addison-Wesley, London, 1984.

[**GoHe77**] S. E. Goodman and S. T. Hedetniemi, *Introduction to the Design and Analysis of Algorithms*, McGraw-Hill, New York, 1977.

[**GoGo78**] C. C. Gotlieb and L. R. Gotlieb, *Data Types and Structures*, Prentice Hall, Englewood Cliffs, NJ, 1978.

[**GrHe85**] R. L. Graham and P. Hell, "On the History of the Minimum Spanning Tree Problem," *Annals of the History of Computing*, 7 (1985), 43-57.

[**GrKnPa94**] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2d ed., Addison-Wesley, Reading, MA, 1994.

[**GrWa77**] J. E .Graver and M .E. Watkins, *Combinatorics with Emphasis on the Theory of Graphs*, Springer-Verlag, New York, 1977.

[**GrSc93**] D. Gries and F. B. Schneider, A *Logical Approach to Discrete Math*, Springer-Verlag, New York, 1993.

[**Gr94**] R. P. Grimaldi, *Discrete and Combinatorial Mathematics*, 3d ed., Addison-Wesley, Reading, MA, 1994.

[**GrYe99**] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, Boca Raton, FL, 1999.

[**Gr97**] J. Gruska, *Foundations of Computing*, International Thomsen Computer Press, London, 1997.

[**Ha60**] P. R. Halmos, *Naive Set Theory*, D. Van Nostrand, New York, 1960.

[**Ha80**] R. W. Hamming, *Coding and Information Theory*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[**Ha69**] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[**HaWr79**] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 5th ed., Oxford University Press, Oxford, England, 1979.

[**Ha87**] D. Harel, Algorithmics, *The Spirit of Computing*, Addison-Wesley, Reading, MA, 1987.

[**He77**] F. Hennie, *Introduction to Computability*, Addison-Wesley, Reading, MA, 1977.

[**He88**] R. Herken, *The Universal Turing Machine*, *A Half-Century Survey*, Oxford Uni-

versity Press, New York, 1988.

[**Ho66**] F. E. Hohn, *Applied Boolean Algebra*, 2d ed., Macmillan, New York, 1966.

[**Ho99**] D. Hofstadter, *Gödel, Escher, Bach: An Internal Golden Braid*, Basic Books, New York, 1999.

[**HoUl79**] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA., 1979.

[**HoMo76**] D. Hopkin and B. Moss, *Automata*, Elsevier, North-Holland, New York, 1976.

[**HoSa82**] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Rockville, MD, 1982.

[**Hu82**] T. C. Hu, *Combinatorial Algorithms*, Addison-Wesley, Reading, MA, 1982.

[**Ka53**] M. Karnaugh, "The Map Method for Synthesis of Combinatorial Logic Circuits," *Transactions of the AIEE*, part I, 72 (1953), 593-599.

[**Kl56**] S. C. Kleene, "Representation of Events by Nerve Nets," in *Automata Studies*, 3-42, Princeton University Press, Princeton, NJ, 1956.

[**Kn77**] D. E. Knuth, "Algorithms," *Scientific American*, 236, no.4 (April 1977), 63-80.

[**Kn97a**] D. E. Knuth, *The Art of Computer Programming, Vol. I: Fundamental Algorithms*, 3d ed., Addison-Wesley, Reading, MA, 1997a.

[**Kn97b**] D. E. Knuth, *The Art of Computer Programming, Vol. II: Seminumerical Algorithms*, 3d ed., Addison-Wesley, Reading, MA, 1997b.

[**Kn98**] D. E. Knuth, *The Art of Computer Programming, Vol. III: Sorting and Searching*, 2d ed., Addison-Wesley, Reading, MA, 1998.

[**Ko86**] Z. Kohavi, *Switching and Finite Automata Theory*, 2d ed., McGraw-Hill, New York, 1986.

[**Kr87**] L. Kronsjö, Algorithms: *Their Complexity and Efficiency*, 2d ed., Wiley, New York, 1987.

[**Kr56**] J. B. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proceedings of the AMS*, 1 (1956), 48-50.

[**Le64**] D. H. Lehmer, "The Machine Tools of Combinatorics," in E.F.Beckenbach (ed.), *Applied Combinatorial Mathematics*, Wiley, New York, 1964.

[**Le77**] W. J. LeVeque, *Fundamentals of Number Theory*, Addison-Wesley, Reading, MA, 1977.

[**LePa81**] H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, Prentice Hall, Englewood Cliffs, NJ, 1981.

[**LiLi81**] Y. Lin and S. Y. T. Lin, *Set Theory with Applications*, 2d ed., Mariner, Tampa, FL, 1981.

[**Li85**] C. L. Liu, *Elements of Discrete Mathematics*, 2d ed., McGraw-Hill, New York, 1985.

[**Li68**] C. L. Liu, *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York,

1968.

[**Lo79**] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland, Amsterdam, 1979.

[**Lu91**] E. Lucas, *Récreations Mathématiques*, Gauthier-Villars, Paris, 1891.

[**M91**] J. C. Martin, *Introduction to Languages and the Theory of Computation*, McGraw-Hill, New York, 1991.

[**Ma93**] H. F. Mattson, Jr., *Discrete Mathematics with Applications*, Wiley, New York, 1993.

[**Mc56**] E. J. McCluskey, Jr., "Minimization of Boolean Functions," *Bell System Technical Journal*, 35 (1956), 1417-1444.

[**Mc90**] J. A. McHugh, *Algorithmic Graph Theory*, Prentice Hall, Englewood Cliffs, NJ, 1990.

[**Mc82**] R. McNaughton, *Elementary Computability*, *Formal Languages*, *and Automata*, Prentice Hall, Englewood Cliffs, NJ, 1982.

[**Me55**] G. H. Mealy, "A Method for Synthesizing Sequential Circuits," *Bell System Technical Journal*. 34 (1955), 1045-1079.

[**Me64**] E. Mendelson, *Introduction to Mathematical Logic*, Van Nostrand Reinhold, New York, 1964.

[**MeOoVa97**] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

[**MiRo91**] J. G. Michaels and K. H. Rosen, *Applications of Discrete Mathematics*, McGraw-Hill, New York, 1991.

[**Mo69**] J. R. Monk, *Introduction to Set Theory*, McGrawHill, New York, 1969.

[**Mo56**] E. F. Moore, "Gedanken-Experiments on Sequen-tial Machines," in *Automata Studies*, 129-153, Princeton University Press, Princeton, NJ, 1956.

[**Ne85**] C. V. Negoita, *Expert Systems and Fuzzy Systems*, Benjamin Cummings, Menlo Park, CA, 1985.

[**Or63**] O. Ore, *Graphs and Their Uses*, Mathematical Association of America, Washington, D.C., 1963.

[**PoSh81**] I. Pohl and A. Shaw, *The Nature of Computation*: *An Introduction to Computer Science*, Computer Science Press, Rockville, MD, 1981.

[**Po54**] G. Pólya, *Mathematics and Plausible Reasoning*, Princeton University Press, Princeton, NJ, 1954.

[**Po57**] G. Pólya, *How to Solve It*, Doubleday, Garden City, NY, 1957.

[**Po62**] G. Pólya, *Mathematical Discovery*, Wiley, New York, 1962.

[**Po83**] G. Pólya, R. E. Tarjan, and D. R. Woods, *Notes on Introductory Combinatorics*, Birkhäuser, Boston, 1983.

[**Pr57**] R. C. Prim, "Shortest Connection Networks and Some Generalizations," *Bell System Technical Journal*, 36 (1957), 1389-1401.

[**PuBr85**] P. W. Purdom, Jr. and C. A. Brown, *The Analysis of Algorithms*, Holt, Riinehart & Winston, New York, 1985.

[**Qu52**] W. V. Quine, "The Problem of Simplifying Truth Functions," *American Mathematical Monthly*, 59 (1952), 521–531.

[**Qu55**] W. V. Quine, "A Way to Simplify Truth Functions," *American Mathematical Monthly*, 62 (1955), 627–631.

[**Ra62**] T. Rado, "On Non-Computable Functions," *Bell System Technical Journal*, (May 1962), 877-844.

[**ReNiDe77**] E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

[**Ri58**] J. Riordan, *An Introduction to Combinatorial Analysis*, Wiley, New York, 1958.

[**Ri68**] J. Riordan, *Combinatorial Identities*, Wiley, New York, 1968.

[**RiShAd78**] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the Association for Computing Machinery*, 31, no.2 (1978), 120–128.

[**Ro76**] F. S. Roberts, *Discrete Mathematics Models*, Prentice Hall, Englewood Cliffs, NJ, 1976.

[**Ro84**] F. S. Roberts, *Applied Combinatorics*, Prentice Hall, Englewood Cliffs, NJ, 1984.

[**Ro86**] E. S. Roberts, *Thinking Recursively*, Wiley, New York, 1986.

[**RoFo80**] D. F. Robinson and L. R. Foulds, *Digraphs: Theory and Techniques*, Gordon and Breach, New York, 1980.

[**Ro84**] J. S. Rohl, *Recursion via Pascal*, Cambridge University Press, Cambridge, England, 1984.

[**Ro93**] K. H. Rosen, *Elementary Number Theory and Its Applications*, 3d ed., Addison-Wesley, Reading, MA, 1993.

[**Ro99**] K. H. Rosen, *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, Boca Raton, FL, 1999.

[**Ro84**] S. Ross, A *First Course in Probability*, 2d ed., Macmillan, New York, 1984.

[**Ro59**] B. Roy, "Transitivité et Connexité," *C.R.Acad.Sci.Paris*, 249 (1959), 216.

[**Ry63**] H. Ryser, *Combinatorial Mathematics*, Mathematical Association of America, Washington, D.C., 1963.

[**SaKa86**] T. L. Saaty and P. C. Kainen, *The Four-Color Problem: Assaults and Conquest*, Dover, New York, 1986.

[**Sa85**] S. Sahni, *Concepts in Discrete Mathematics*, Camelot, Minneapolis, 1985.

[**SePi89**] J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[**Se88**] R. Sedgewick, *Algorithms*, 2d ed., Addison-Wesley, Reading, MA, 1988.

[**SiHs88**] H. J. Siegel and W. T. Hsu, "Interconnection Networks," in *Computer Architectures*, V.M.Milutinovic (ed.), North-Holland, New York, 1988, pp.225–264.

[Si66] A. Sinkov, *Elementary Cryptanalysis*, Mathematical Association of America, Washington, D.C., 1966.

[SlPl95] N. J. A. Sloane and S. Plouffe, *The Encylopedia of Integer Sequences*, Academic Press, New York, 1995.

[So61] I. S. Sominskii, *Method of Mathematical Induction*, Blaisdell, New York, 1961.

[Sp94] M. Spivak, *Calculus*, 3d ed., Publish or Perish, Wilmington, DE, 1994.

[StMc77] D. Stanatand and D. F. McAllister, *Discrete Mathematics in Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1997.

[St78] H. M. Stark, *An Introduction to Number Theory*, MIT Press, Cambrdge, MA, 1978.

[St95] D. R. Stinson, *Cryptography*, *Theory and Practice*, CRC Press, Boca Raton, FL, 1995.

[St94] P. K. Stockmeyer, "Variations on the Four-Post Tower of Hanoi Puzzle," *Congressus Numerantrum* 102 (1994), 3-12.

[St74] R. R. Stoll, *Sets Logic*, *and Axiomatic Theories*, 2d ed., W.H.Freeman, San Francisco, 1974.

[St88] G. W. Strang, *Linear Algebra and Its Applications*, 3d ed., Harcourt Brace Jovanovich, San Diego, 1988.

[Su87] P. Suppes, *Introduction to Logic*, D.Van Nostrand, Princeton, NJ, 1987.

[Ta83] R. E. Tarjan, *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, 1983.

[ThFi96] G. B. Thomas and R. L. Finney, *Calculus and Analytic Geometry*, 9th ed., Addison-Wesley, Reading, MA, 1996.

[TrMa75] J. P. Tremblay and R. P. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw-Hill, New York, 1975.

[Tu85] A. Tucker, *Applied Combinatorics*, 2d ed., Wiley, New York, 1985.

[Ve52] E. W. Veitch, "A Chart Method for Simplifying Truth Functions," *Proceedings of the ACM*, (1952), 127-133.

[Vi71] N. Y. Vilenkin, *Combinatorics*, Academic Press, New York, 1971.

[Wa80] M. Wand, *Induction*, *Recursion*, *and Programming*, North-Holland, New York, 1980.

[Wa62] S. Warshall, "A Theorem on Boolean Matrices," *Journal of the ACM*, 9 (1962), 11-12.

[Wi86] Herbert S. Wilf, *Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, NJ, 1986.

[Wi85] S. G. Williamson, *Combinatorics for Computer Science*, Computer Science Press, Rockville, MD, 1985.

[Wi85] R. J. Wilson, *Introduction to Graph Theory*, 3d ed., Longman, Essex, England, 1985.

[**WiBe79**] R. J. Wilson and L. W. Beineke, *Applications of Graph Theory*, Academic Press, London, 1979.

[**WiWa90**] R. J. Wilson and J. J. Watkins, *Graphs*, *An Introductory Approach*, Wiley, New York, 1990.

[**Wi76**] N. Wirth, *Algorithms + Data Structures = Programs*, Prentice Hall, Englewood Cliffs, NJ, 1976.

[**Wi84**] N. Wirth, "Data Structures and Algorithms," *Scientific American*, 251 (September 1984), 60-69.

[**Wo87**] D. Wood, *Theory of Computation*, Harper & Row, New York, 1987.

[**Zi91**] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 2d ed., Kluwer, Boston, 1991.